Network Inference from Co-Occurrences

Michael G. Rabbat, Mário A. T. Figueiredo, and Robert D. Nowak

February 1, 2008

Abstract

The discovery of network structures is a fundamental problem in arising in numerous fields of science and technology, communication systems, biology, sociology and neuroscience. Unfortunately, it is often difficult to obtain data that directly reveals network structure, and so one must infer a network from incomplete data. This paper considers inferring network structure from "co-occurrence" data; observations that identify which network components (e.g., switches, routers, genes) carry each transmission but does not indicate the order in which they handle the transmissions. Without order information, there is an exponential number of feasible networks that are compatible with the observed data. Yet, the basic physical principles underlying most networks strongly suggest that all feasible networks are not equally likely. In particular, network elements that co-occur in many observations are probably closely connected. We model the co-occurrence observations as independent realizations of a random walk on the underlying graph, subjected to a random permutation which accounts for the lack of order information. Treating the permutations as missing data, we derive an exact *expectation-maximization* (EM) algorithm for estimating the random walk parameters. The model and EM algorithm significantly simplify the problem, but the computational complexity of the reconstruction process does grow exponentially in the length of the longest transmission path. For large networks the exact E-step may be computationally intractable, and so we also propose an efficient *Monte Carlo EM* (MCEM) algorithm, based on importance sampling, and derive conditions which ensure convergence of the algorithm with high probability. Remarkably, the MCEM maintains the desirable properties of the exact EM algorithm and reduces the complexity of each iteration

M.G. Rabbat and R.D. Nowak are with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI, 53706. Email: `rabbat@cae.wisc.edu, nowak@engr.wisc.edu`.

M.A.T. Figueiredo is with *Instituto de Telecomunicações* and the Department of Electrical and Computer Engineering, *Instituto Superior Técnico*, Lisboa, Portugal. Email: `mario.figueiredo@lx.it.pt`.

to polynomial-time. Simulations and experiments with Internet measurements demonstrate the promise of this approach.

1 Network Inference and Co-Occurrence Observations

The study of complex networked systems is an emerging field, impacting nearly every area of engineering and science, including the important domains of communication systems, biology, sociology, and cognitive science. The analysis of communication networks enables a better understanding of routing, transmission patterns, and information flow [6, 21]. The analysis of biological networks provides insight into the functional roles played by different genes, proteins, and metabolites in biological systems [13, 19]. The analysis of social networks contributes to a deeper understanding of interactions, dynamics, and the structure of organizations [18, 25]. The analysis of functional connectivity networks in the brain is necessary for the understanding of couplings and interactions between different neuronal colonies [1, 23, 24]. Obtaining or inferring the structure of networks from experimental data precedes any such analysis and is thus a basic and fundamental task, critical to many applications.

Unfortunately, measurements which directly reveal network structure are often beyond experimental capabilities or are excessively expensive. This paper considers inferring network structure from observations that identify which network components (e.g., switches, routers, genes) carry each transmission but does not indicate the order in which they handle the transmissions. Mathematically, the underlying network structure can be represented as a directed graph, and the vertices involved in each transmission form a connected subgraph. The observations only reflect which subset of vertices are involved or "occur" in each transmission; not their inter-connectivity. We refer to such observations as *co-occurrences*. Co-occurrence observations arise naturally in each of the application areas mentioned above.

Transmissions over telecommunication networks are carried by links and routers/switches which form a path between the source and terminal nodes. In some cases, it is impossible to directly observe the order in which the routers/switches handle each transmission, since sensors are geographically distributed, making precise time-synchronization impractical. The so-called *internally-sensed*

network tomography problem specifically aims at recovering network structure from unordered lists of network elements along transmission paths [21].

Biological signal transduction networks describe fundamental cell functions such as growth, metabolism, differentiation, and apoptosis (disintegration) [19]. Although it is possible to test for individual, localized interactions between genes pairs, such experiments are expensive and time-consuming. High-throughput measurement techniques such as microarrays have successfully been used to identify the components of different signal transduction pathways [28]. However, microarray data only reflects order information at a very coarse, unreliable level. Developing computational techniques for inferring pathway orders is an active research area [16].

Co-occurrence or transactional data also appears in the context of social networks, *e.g.*, by considering which academic papers are co-cited by another paper, which web pages are linked to or from another web page, or which people were diagnosed with a common disease on the same day. Such measurements are readily available, but do not necessarily reflect the temporal or other natural order of occurrence. Researchers in this area have considered the problems of reconstructing networks from co-occurrence data and of using the inferred network to predict potential future co-occurrences [14].

Functional magnetic resonance imaging (fMRI) provides a mechanism for measuring activity in the brain with high spatial resolution. By observing which regions of the brain co-activate while a patient is performing different tasks we can obtain multiple co-occurrence observations. Although fMRI offers high spatial resolution it has limited temporal resolution, making it impractical to obtain complete order information. Magnetoencephalography and electroencephalography measure activity in the brain with higher temporal resolution but only provide coarse spatial resolution, and thus may not allow one to determine precisely which functional regions are active during a given task. Existing techniques for obtaining functional co-activation networks either involve brute-force measurement or use crude correlation methods (see [24] and references therein).

In this article we focus on observations arising from transmissions in a network. Specifically, each co-occurrence observation corresponds to a path[1] through the network. We observe the vertices

---

[1] Throughout this paper a "path" refers to a sequence of vertices $(x_1, x_2, \ldots, x_N)$ such that there is an edge between each adjacent pair of vertices, x_{i-1} and x_i, and no node appears more than once in the sequence.

comprising each path but not the order in which they appear along the path. In certain applications the endpoints (source and destination) of the path may also be observed.

Our goal is to identify which pairs of vertices are directly connected via an edge, thereby learning the structure of the network. A *feasible graph* is one which agrees with the observations; *i.e.*, a graph which contains a directed path through the vertices in each co-occurrence observation. Given a collection of co-occurrence observations a feasible graph is easily constructed by assigning an order – any order, in fact – to the vertices in each observation, and then inserting directed edges between vertices which are adjacent in the assigned order. In light of the many possible orders for each co-occurrence observation, the number of feasible topologies grows exponentially in the number and size of observations. Without additional assumptions, side information, or prior knowledge, there is no reason to prefer one feasible topology over the others.

Previous work on related problems has involved heuristics using frequencies of co-occurrence either to assign an order to each path [21] or to approximate the probability of transitioning from one vertex to another [14]. These approaches make stringent assumptions and sacrifice flexibility in order to achieve computational tractability and systematically identify a unique solution. The *frequency method* introduced in [21] is based on a model where paths from a particular source or to a particular destination form a tree. This model coincides with the shortest-path routing policy. When the network provides multiple paths between the same pair of endpoints (*e.g.*, for load-balancing) the algorithm may fail. The *cGraph* algorithm of Kubica *et al.* [14] inserts weighted edges between every pair of vertices which co-occur in some observation. This approach produces solutions which are typically much denser than desired. Because both of these methods are based on heuristics, the results they produce are not easily interpreted. Also, these heuristics do not readily lend themselves to incorporating side information. A different approach, introduced by Justice and Hero in [12], involves averaging over an ensemble of feasible topologies sampled uniformly from the feasible set. In general there is an enormous number of feasible topologies (exponential in the problem dimensions) exhibiting a wide variety of characteristics, and it is not clear that an average of feasible topologies will be optimal in any sense. These observations have collectively motivated our development of a more general approach to network reconstruction which we simply

term *network inference from co-occurrences*, or NICO for short.

Our approach is based on a generative model where paths are realizations of a random walk on the underlying graph. A co-occurrence observation is obtained by randomly shuffling each path to account for our lack of observed order information. Based on this model, network inference reduces to estimating the parameters governing the random walk. Then, these parameter estimates determine the most likely order for each co-occurrence.

The following interpretation motivates our shuffled random walk model. Imagine sitting at a particular vertex in the network and observing a series of transmissions pass by. This vertex is only connected to a handful of other vertices in the network, so regardless of its final destination, a transmission arriving at this vertex must pass through one of the neighboring vertices next. Over a period of time, we could record how many arriving transmissions are passed to each neighbor, and then calculate the empirical probabilities of transmissions to each of the neighbors. Obtaining such probabilities at each vertex would provide a tremendous amount of information about the network, but unfortunately co-occurrence observations do not directly reveal them and we therefore face a challenging inverse problem. This paper develops a formal framework for estimating local transition probabilities from a collection of co-occurrence observations, without making any additional assumptions about routing behavior or properties of the underlying network structure. Experimental results on simulated topologies indicate that good performance is obtained for a variety of operating conditions.

It is also worth mentioning that the approach discussed in this paper differs considerably from that of learning the structure of a directed graphical model or Bayesian network, a graph where nodes correspond to random variables and edges indicate conditional independence relationships [8,9]. A typical aim of graphical modelling is to find a graph corresponding to a factorization of a high-dimensional distribution which predicts the observations well. In turn, these probabilistic models do not directly reflect physical structures, and applying such an approach in the context of this problem would ignore physical constraints inherent to the observations: that co-occurring vertices must lie along a path in the network.

The rest of the paper is organized as follows. In Section 2 we introduce notation and formulate

the problem setup. Section 3 reviews the standard approach to estimating the parameters of a random walk when fully observed (ordered) samples are available and presents an EM algorithm for estimating random walk parameters from shuffled observations. A Monte Carlo variant of the EM algorithm is described in Section 4 for situations where long transmission paths make the exact E-step computation prohibitive. Section 5 analyzes convergence of the Monte Carlo EM algorithm. Simulation results are presented in Section 6 and the paper is concluded in Section 7, where ongoing work is also briefly described.

2   Problem Formulation

We model the network as a simple directed graph $G = (S, E)$, where $S = \{1, 2, \ldots, |S|\}$ is the set of vertices/nodes and $E \subseteq S \times S$ is the set of edges. The number of nodes, $|S|$, is considered known, so network inference amounts to determining the adjacency structure of the graph; that is, identifying whether or not $(i, j) \in E$, for every pair of vertices $(i, j) \in S \times S$.

A co-occurrence observation, $\mathbf{y} \subset S$, is a subset of vertices in the graph which simultaneously "occur" when a particular stimulus is presented to the network. For example, when a transmission is made over a communication network, a subset of routers and switches carry the transmission from the source to the destination. This activated subset corresponds to a co-occurrence observation, with the stimulus being a transmission between that particular source-destination pair. By repeating this procedure T times with different stimuli we obtain observations, $\mathcal{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(T)}\}$, where $\mathbf{y}^{(m)} = (y_1^{(m)}, y_2^{(m)}, \ldots, y_{N_m}^{(m)})$ is a length-N_m co-occurrence, indexed in an arbitrary order.

A directed graph $G = (S, E)$ is said to be feasible with respect to observations $\mathcal{Y}$ if for each co-occurrence $\mathbf{y}^{(m)} \in \mathcal{Y}$ there exists an ordered path $\mathbf{z}^{(m)} = (z_1^{(m)}, z_2^{(m)}, \ldots, z_{N_m}^{(m)})$ and a permutation $\boldsymbol{\tau}^{(m)} = (\tau_1^{(m)}, \ldots, \tau_{N_m}^{(m)})$ such that $z_t^{(m)} = y_{\tau_t^{(m)}}^{(m)}$ for each t, and there is an edge from z_{t-1} to z_t in the graph for $t = 2, \ldots, N_m$, that is, $(z_{t-1}, z_t) \in E$.

Notice that if we observed ordered paths then network inference would be trivial. We would begin with an empty graph $G_0 = (S, E)$ with $E = \{\}$. Then, for each ordered observation $\mathbf{z}^{(m)}$ we would update the set of edges via $E \leftarrow E \cup (z_{t-1}^{(m)}, z_t^{(m)})$ for $t = 2, \ldots, N_m$. Similarly, if we observed the correct permutation $\boldsymbol{\tau}^{(m)}$ along with each co-occurrence $\mathbf{y}^{(m)}$, we could invert the permutation

to recover ordered observations and apply the same procedure.

In practice we do not make ordered observations nor do we have access to the correct permutations. However, we can obtain a feasible reconstruction by associating *any* permutation (of the appropriate length) with each co-occurrence, and then following the procedure described above. There are $N_m!$ ways to permute the elements of $\mathbf{y}^{(m)}$, so there may be as many as $\prod_{m=1}^{T} N_m!$ feasible reconstructions. Clearly, for large N_m and T this is a huge set to search over. Moreover, without making additional assumptions, or adopting some additional criteria, there is no reason to prefer one feasible reconstruction over another.

Physical principles governing the development of many natural and man-made networks suggest that not all feasible networks are equally plausible. Intuitively, if two or more vertices appear collectively in multiple co-occurrences, we expect that their order is probably the same in the corresponding paths. Likewise, we expect that most vertices will only be directly connected to a small fraction of the other vertices. Based on this intuition we propose the following probabilistic model. First, we model the unobserved, ordered paths, $\mathbf{z}^{(m)}$, as independent samples of a first-order Markov chain. The Markov chain is parameterized by an initial state distribution $\boldsymbol{\pi} \in [0,1]^{|S|}$ where $\pi_i = P[z_1 = i]$, and a probability transition matrix, $\mathbf{A} \in [0,1]^{|S| \times |S|}$, where $A_{i,j} = P[z_t = j | z_{t-1} = i]$. Of course, these parameters must satisfy the normalization constraints

$$\sum_{i=1}^{|S|} \pi_i = 1 \quad \text{and} \quad \sum_{j=1}^{|S|} A_{i,j} = 1, \qquad \text{for each } i = 1, \ldots, |S|. \tag{1}$$

In addition, we assume that the support of the transition matrix is determined by the adjacency structure of the underlying network; *i.e.*, $A_{i,j} > 0$ if and only if $(i,j) \in E$.

A co-occurrence observation, $\mathbf{y}$, is generated by shuffling the elements of an ordered Markov chain sample, $\mathbf{z} = (z_1, \ldots, z_N)$, via a permutation $\boldsymbol{\tau}$ drawn uniformly from Ψ_N, the collection of all permutations of N elements. Thus, for each $t = 1, \ldots, N$, $z_t = y_{\tau_t}$. We assume that $\boldsymbol{\tau}$ is independent of the Markov chain sample, $\mathbf{z}$. Based on this model, we can write the likelihood of a

co-occurrence observation $\mathbf{y}$ conditioned on the permutation $\boldsymbol{\tau}$ as

$$P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}] \quad = \quad \pi_{y_{\tau_1}} \prod_{t=2}^{N} A_{y_{\tau_{t-1}}, y_{\tau_t}}. \tag{2}$$

Since $P[\boldsymbol{\tau}] = 1/(N!)$, for any $\boldsymbol{\tau} \in \Psi_N$, marginalization over all permutations leads to

$$P[\mathbf{y}|\mathbf{A}, \boldsymbol{\pi}] = \frac{1}{N!} \sum_{\boldsymbol{\tau} \in \Psi_N} P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}]. \tag{3}$$

Finally, assuming that co-occurrence observations are independent, and taking the logarithm, gives

$$\log P[\mathcal{Y}|\mathbf{A}, \boldsymbol{\pi}] \quad = \quad \sum_{m=1}^{T} \left[\log \left(\sum_{\boldsymbol{\tau} \in \Psi_{N_m}} P[\mathbf{y}^{(m)}|\boldsymbol{\tau}^{(m)}, \mathbf{A}, \boldsymbol{\pi}] \right) - \log(N_m!) \right]. \tag{4}$$

Under this model, network inference consists in computing the maximum likelihood (ML) estimates,

$$(\widehat{\mathbf{A}}_{\mathrm{ML}}, \widehat{\boldsymbol{\pi}}_{\mathrm{ML}}) = \arg \max_{\mathbf{A}, \boldsymbol{\pi}} \log P[\mathcal{Y}|\mathbf{A}, \boldsymbol{\pi}]. \tag{5}$$

With the ML estimates in hand, we may determine the most likely permutation for each co-occurrence observation according to $(\mathbf{A}_{\mathrm{ML}}, \boldsymbol{\pi}_{\mathrm{ML}})$, and obtain a feasible reconstruction using our procedure for ordered observations described above.

For non-trivial observations, $\log P[\mathcal{Y}|\mathbf{A}, \boldsymbol{\pi}]$ is a complicated, non-concave function of $(\mathbf{A}, \boldsymbol{\pi})$, so solving (5) is not a simple task. In the next section, we derive a EM algorithm for solving this optimization problem, by treating the set of permutations, $\mathcal{T} = \{\boldsymbol{\tau}^{(1)}, ..., \boldsymbol{\tau}^{(T)}\}$, shuffling the paths, as missing data.

3 An EM Algorithm for Estimating Markov Chain Parameters from Shuffled Observations

3.1 Fully Observed Markov Chains: Notation and Estimation

Let $\mathcal{Z} = \{\mathbf{z}^{(1)}, ..., \mathbf{z}^{(T)}\}$ be a set of sample paths, $\mathbf{z}^{(m)} = (z_1^{(m)}, ..., z_{N_m}^{(m)})$, independently generated by a Markov chain with transition matrix $\mathbf{A}$ and initial state distribution $\boldsymbol{\pi}$ (see (1)). For later use, it is convenient to introduce the equivalent binary representation $\mathbf{w}^{(m)} = (\mathbf{w}_1^{(m)}, ..., \mathbf{w}_{N_m}^{(m)})$, for each sample $\mathbf{z}^{(m)}$, defined as follows: $\mathbf{w}_t^{(m)} = (w_{t,1}^{(m)}, ..., w_{t,|S|}^{(m)}) \in \{0,1\}^{|S|}$, with $(w_{t,i}^{(m)} = 1) \Leftrightarrow (z_t^{(m)} = i)$; of course, one and only one entry of each vector $\mathbf{w}_t^{(m)}$ equals 1. Finally, let $\mathcal{W} = \{\mathbf{w}^{(1)}, ..., \mathbf{w}^{(T)}\}$, which contains the exact same information as $\mathcal{Z}$. With this notation, we can write

$$
\begin{aligned}
\log P[\mathcal{W}|\mathbf{A}, \boldsymbol{\pi}] &= \sum_{m=1}^{T}\sum_{i=1}^{|S|} w_{1,i}^{(m)} \log \pi_i + \sum_{m=1}^{T}\sum_{t=2}^{N_m}\sum_{i=1}^{|S|}\sum_{j=1}^{|S|} w_{t-1,i}^{(m)} w_{t,j}^{(m)} \log A_{i,j}. \\
&= \sum_{i=1}^{|S|} \log \pi_i \sum_{m=1}^{T} w_{1,i}^{(m)} + \sum_{i=1}^{|S|}\sum_{j=1}^{|S|} \log A_{i,j} \sum_{m=1}^{T}\sum_{t=2}^{N_m} w_{t-1,i}^{(m)} w_{t,j}^{(m)}.
\end{aligned}
$$

Maximum likelihood estimates of $\boldsymbol{\pi}$ and $\mathbf{A}$ can be obtained from $\mathcal{W}$ my maximizing $\log P[\mathcal{W}|\mathbf{A}, \boldsymbol{\pi}]$ under the constraints in (1); the solution is well known,

$$
\widehat{A}_{i,j} = \frac{\displaystyle\sum_{m=1}^{T}\sum_{t=2}^{N_m} w_{t-1,i}^{(m)} w_{t,j}^{(m)}}{\displaystyle\sum_{j=1}^{|S|}\sum_{m=1}^{T}\sum_{t=2}^{N_m} w_{t-1,i}^{(m)} w_{t,j}^{(m)}}, \qquad \text{and} \qquad \widehat{\pi}_i = \frac{1}{T}\sum_{m=1}^{T} w_{1,i}^{(m)}. \tag{6}
$$

3.2 Shufflings, Permutations, and the EM Algorithm

To address the case where we have a set of co-occurrences $\mathcal{Y} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(T)}\}$, not ordered samples, we defined the equivalent binary representation $\mathcal{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}\}$ for $\mathcal{Y}$ in a similar way as above: $\mathbf{x}^{(m)} = (\mathbf{x}_1^{(m)}, ..., \mathbf{x}_{N_m}^{(m)})$, where $\mathbf{x}_t^{(m)} = (x_{t,1}^{(m)}, ..., x_{t,|S|}^{(m)}) \in \{0,1\}^{|S|}$, with $(x_{t,i}^{(m)} = 1) \Leftrightarrow (y_t^{(m)} = i)$.

Rather than using $\boldsymbol{\tau}^{(m)} = (\tau_1^{(m)}, \ldots, \tau_N^{(m)})$ to denote the mth permutation/shuffling, we introduce a more convenient (binary) representation; each shuffling is represented by a *permutation*

$matrix^2$, which we will term *shuffling matrix*. Let the shuffling matrix for sequence m be denoted as $\mathbf{r}^{(m)}$ so that $(r_{t,t'}^{(m)} = 1) \Leftrightarrow (\tau_t = t') \Leftrightarrow (\mathbf{x}_{t'}^{(m)} = \mathbf{w}_t^{(m)}) \Leftrightarrow (y_{t'}^{(m)} = z_t^{(m)})$. Given both $\mathbf{r}^{(m)}$ and $\mathbf{x}^{(m)}$, we recover the unshuffled sequence $\mathbf{w}^{(m)}$ by applying (using $0^0 = 1$)

$$w_{t,i}^{(m)} = \prod_{t'=1}^{N_m} \left(x_{t',i}^{(m)} \right)^{r_{t,t'}^{(m)}}. \tag{7}$$

Let $\mathcal{R} = \{\mathbf{r}^{(1)}, ..., \mathbf{r}^{(T)}\}$ be the collection of shuffling matrices that allow recovering the underlying ordered paths $\mathcal{W} = \{\mathbf{w}^{(1)}, ..., \mathbf{w}^{(T)}\}$ from the corresponding shuffled co-occurrences $\mathcal{X} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(T)}\}$. We can write the complete log-likelihood $\log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}]$ as follows:

$$
\begin{aligned}
\log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}] &= \log P[\mathcal{X} | \mathcal{R}, \mathbf{A}, \boldsymbol{\pi}] + \log p[\mathcal{R}] \\
&= \sum_{m=1}^{T} \log P[\mathbf{x}^{(m)} | \mathbf{r}^{(m)}, \mathbf{A}, \boldsymbol{\pi}] + \log p[\mathcal{R}] \\
&= \sum_{m=1}^{T} \sum_{t=2}^{N_m} \sum_{t'=1}^{N_m} \sum_{t''=1}^{N_m} \sum_{i,j=1}^{|S|} r_{t,t'}^{(m)}\, r_{t-1,t''}^{(m)}\, x_{t'',i}^{(m)}\, x_{t',j}^{(m)}\, \log A_{i,j} \\
&\quad + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \sum_{i=1}^{|S|} r_{1,t'}^{(m)}\, x_{t',i}^{(m)}\, \log \pi_i + \log p[\mathcal{R}],
\end{aligned}
$$

where $p[\mathcal{R}]$ is the probability of the set of shufflings $\mathcal{R}$, which we assume constant.

To estimate $\mathbf{A}$ and $\boldsymbol{\pi}$ from $\mathcal{X}$, we treat $\mathcal{R}$ as missing data, opening the door to the use of the EM algorithm. Notice that if we had the complete data $(\mathcal{X}, \mathcal{R})$, we could recover $\mathcal{W}$ via (7) and obtain the closed-form estimates (6). The EM algorithm proceeds by computing the expected value of $\log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}]$ (w.r.t. $\mathcal{R}$), conditioned on the observations and on the current model estimate $(\mathbf{A}^k, \boldsymbol{\pi}^k)$ (the E-step),

$$Q\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right) = E\left[\log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}] \,\Big|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k\right]. \tag{10}$$

---

[2] A matrix with one and only one "1" in each row and each column.

The model parameter estimates are then updated as follows (the M-step):

$$\left(\mathbf{A}^{k+1}, \boldsymbol{\pi}^{k+1} \right) = \arg \max_{\mathbf{A}, \boldsymbol{\pi}} \; Q \left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k \right). \tag{11}$$

These two steps are repeated cyclically until a convergence criterion is met.

3.3 The E-step

3.3.1 Sufficient statistics

Rearranging (9), and dropping $\log P[\mathcal{R}]$ (assumed constant), we can write

$$
\begin{aligned}
\log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}] \;\; \propto \;\; & \sum_{m=1}^{T} \sum_{i,j=1}^{|S|} \sum_{t',t''=1}^{N_m} \sum_{t=2}^{N_m} r_{t,t'}^{(m)} r_{t-1,t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} \log A_{i,j} \\
& + \sum_{m=1}^{T} \sum_{i=1}^{|S|} \sum_{t'=1}^{N_m} r_{1,t'}^{(m)} x_{t',i}^{(m)} \log \pi_i.
\end{aligned}
\tag{12}
$$

revealing that $\log P[\mathcal{X}, \mathcal{R} | \mathbf{A}, \boldsymbol{\pi}]$ is linear with respect to simple functions of $\mathcal{R}$:

- the first row of each $\mathbf{r}^{(m)}$: $r_{1,t'}^{(m)}$, for $m = 1, \dots, T$ and $t' = 1, \dots, N_m$;

- sums of transition indicators: $\alpha_{t',t''}^{(m)} \equiv \sum_{t=2}^{N_m} r_{t,t'}^{(m)} \, r_{t-1,t''}^{(m)}$, for $m = 1, \dots, T$, and $t', t'' = 1, \dots, N_m$.

Since expectations commute with linear functions, the E-step reduces to computing the conditional expectations of $r_{t,t'}^{(m)}$ and $\alpha_{t',t''}^{(m)}$ (denoted $\bar{r}_{1,t'}^{(m)}$ and $\bar{\alpha}_{t',t''}^{(m)}$, respectively) and plugging them into the complete log-likelihood. Noticing that the $r_{t,t'}^{(m)}$ and $\alpha_{t',t''}^{(m)}$ are binary (in $\{0, 1\}$), yields

$$
\begin{aligned}
\bar{r}_{1,t'}^{(m)} \;\; &= \;\; E \left[r_{1,t'}^{(m)} \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k \right] \;\; = \;\; P \left[r_{1,t'}^{(m)} = 1 \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k \right] \tag{13} \\
\bar{\alpha}_{t',t''}^{(m)} \;\; &= \;\; E \left[\alpha_{t',t''}^{(m)} \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k \right] \;\; = \;\; P \left[\alpha_{t',t''}^{(m)} = 1 \,\middle|\, \mathcal{X}, \mathbf{A}^k, \boldsymbol{\pi}^k \right]. \tag{14}
\end{aligned}
$$

Finally, $Q \left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k \right)$ is obtained simply by plugging $\bar{r}_{1,t'}^{(m)}$ and $\bar{\alpha}_{t',t''}^{(m)}$ in the places of $r_{1,t'}^{(m)}$ and $\sum_{t=2}^{N_m} r_{t,t'}^{(m)} r_{t-1,t''}^{(m)}$, respectively, in (12).

3.3.2 Computing $\bar{r}_{1,t'}^{(m)}$

Since the permutations are (a priori) equiprobable, we have $P[\mathbf{r}] = 1/(N_m!)$ (for $\mathbf{r} \in \Psi_{N_m}$), $P[r_{1,t'}^{(m)} = 1] = ((N_m - 1)!/N_m!) = 1/N_m$, and $P[\mathbf{r}|r_{1,t'}^{(m)} = 1] = 1/((N_m - 1)!)$. Using these facts, together with the mutual independence among the several sequences, and Bayes law, yields

$$
\begin{aligned}
\bar{r}_{1,t'}^{(m)} &= P\left[r_{1,t'}^{(m)} = 1 \,\middle|\, \mathbf{x}^{(m)}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] \\
&= \frac{P\left[\mathbf{x}^{(m)} \middle| r_{1,t'}^{(m)} = 1, \mathbf{A}^k, \boldsymbol{\pi}^k\right] P\left[r_{1,t'}^{(m)} = 1\right]}{P\left[\mathbf{x}^{(m)} \middle| \mathbf{A}^k, \boldsymbol{\pi}^k\right]} \\
&= \frac{\displaystyle\sum_{\mathbf{r} \in \Psi_{N_m}: \, r_{1,t'} = 1} P\left[\mathbf{x}^{(m)} \middle| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\right]}{\displaystyle\sum_{\mathbf{r} \in \Psi_{N_m}} P\left[\mathbf{x}^{(m)} \middle| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\right]},
\end{aligned}
\tag{15}
$$

where each term $P\left[\mathbf{x}^{(m)} \middle| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\right]$ is easily computed after using $\mathbf{r}$ to unshuffle $\mathbf{x}^{(m)}$:

$$
P\left[\mathbf{x}^{(m)} \middle| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] = P\left[\mathbf{y}^{(m)} \middle| \boldsymbol{\tau}, \mathbf{A}^k, \boldsymbol{\pi}^k\right] = \pi_{y_{\tau_1}^{(m)}}^k \prod_{t=2}^{N_m} A_{y_{\tau_{t-1}}^{(m)}, y_{\tau_t}^{(m)}}^k .
$$

Denoting the numerator of (15) as $\gamma_{t'}^{(m)}$ we have a more compact expression

$$
\bar{r}_{1,t'}^{(m)} = \frac{\gamma_{t'}^{(m)}}{\displaystyle\sum_{t'=1}^{N_m} \gamma_{t'}^{(m)}} .
$$

3.3.3 Computing $\bar{\alpha}_{t',t''}^{(m)}$

The computation of $\bar{\alpha}_{t',t''}^{(m)}$ follows a similar path as that of $\bar{r}_{1,t'}^{(m)}$; since all permutations are equiprobable, $P[r_{t,t'}^{(m)} r_{t-1,t''}^{(m)} = 1] = (N_m - 2)!/(N_m!)$ and $P[\mathbf{r}|r_{t,t'}^{(m)} r_{t-1,t''}^{(m)} = 1] = 1/((N_m - 2)!)$, thus

$$
\begin{aligned}
\bar{\alpha}_{t',t''}^{(m)} &= \sum_{t=2}^{N_m} P\big[r_{t,t'}^{(m)} r_{t-1,t''}^{(m)} = 1 \big| \mathbf{x}^{(m)}, \mathbf{A}^k, \boldsymbol{\pi}^k\big] \\
&= \sum_{t=2}^{N_m} \frac{P\big[\mathbf{x}^{(m)} \big| r_{t,t'}^{(m)} r_{t-1,t''}^{(m)} = 1, \mathbf{A}^k, \boldsymbol{\pi}^k\big] \; P\big[r_{t,t'}^{(m)} r_{t-1,t''}^{(m)} = 1\big]}{P\big[\mathbf{x}^{(m)} \big| \mathbf{A}^k, \boldsymbol{\pi}^k\big]} \\
&= \sum_{t=2}^{N_m} \frac{\left(\dfrac{1}{(N_m - 2)!} \displaystyle\sum_{\mathbf{r} \in \Psi_{N_m} : r_{t,t'} r_{t-1,t''}=1} P\big[\mathbf{x}^{(m)} \big| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big] \right) \left(\dfrac{(N_m - 2)!}{N_m!} \right)}{\dfrac{1}{N_m!} \displaystyle\sum_{\mathbf{r} \in \Psi_{N_m}} P\big[\mathbf{x}^{(m)} \big| \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k\big]} \\
&= \frac{\displaystyle\sum_{\mathbf{r} \in \Psi_{N_m}} P[\mathbf{x}^{(m)} | \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k] \displaystyle\sum_{t=2}^{N_m} r_{t,t'} r_{t-1,t''}}{\displaystyle\sum_{\mathbf{r} \in \Psi_{N_m}} P[\mathbf{x}^{(m)} | \mathbf{r}, \mathbf{A}^k, \boldsymbol{\pi}^k]}.
\end{aligned}
\tag{16}
$$

Denoting the numerator of (16) as $\gamma_{t',t''}^{(m)}$, we finally have

$$
\bar{\alpha}_{t',t''}^{(m)} = \frac{\gamma_{t',t''}^{(m)}}{\displaystyle\sum_{t'=1}^{N_m} \gamma_{t',t''}^{(m)}}.
\tag{17}
$$

For the mth observation, the statistics $\{\bar{r}_{1,t'}^{(m)}\}$ and $\{\bar{\alpha}_{t,t',t''}^{(m)}\}$ have an $O(N_m^2)$ memory cost ($N_m^2 - N_m$ transition statistics and N_m initial state statistics). These quantities can be computed via the summary statistics $\{\gamma_{t'}^{(m)}\}$ and $\{\gamma_{t',t''}^{(m)}\}$, using the same memory needed to store $\{\bar{r}_{1,t'}^{(m)}\}$ and $\{\bar{\alpha}_{t,t',t''}\}$, in $O(N_m!)$ operations (the number of all permutations for a length N_m observation). For large N_m, this is a heavy load; Section 4 describes a sampling approach for computing approximations to $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t',t''}$.

3.4 The M-step

Recall that the function $Q\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right)$ is obtained by plugging $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t',t''}$ in the places of $r_{1,t'}^{(m)}$ and $\sum_{t=2}^{N_m} r_{t,t'}^{(m)} r_{t-1,t''}^{(m)}$, respectively, in (12). Maximization w.r.t. $\mathbf{A}$ and $\boldsymbol{\pi}$, under the constraints in (1), leads to following simple update equations:

$$A_{i,j}^{k+1} = \frac{\displaystyle\sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)}}{\displaystyle\sum_{j=1}^{|S|} \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)}} \quad \text{and} \quad \pi_i^{k+1} = \frac{\displaystyle\sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)}}{\displaystyle\sum_{i=1}^{|S|} \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)}}. \tag{18}$$

3.5 Handling Known Endpoints

In some applications, (one or both of) the endpoints of each path are known and only the internal nodes are shuffled. This is the case in communication networks (*i.e.*, internally-sensed network tomography), since the sources and destinations are known, but not the connectivity within the network. In estimation of biological networks (signal transduction pathways), a physical stimulus (*e.g.*, hypotonic shock) causes a sequence of protein interactions, resulting in another observable physical response (*e.g.*, a change in cell wall structure); in this case, the stimulus and response act as fixed endpoints, our goal is to infer the order of the sequence of protein interactions.

Observe that knowledge of the endpoints of each path imposes the constraints

$$r_{1,1}^{(m)} = 1 \quad \text{and} \quad r_{N_m,N_m}^{(m)} = 1.$$

Under the first constraint, estimates of the initial state probabilities are simply given by

$$\widehat{\pi}_i = \frac{1}{T} \sum_{m=1}^{T} x_{1,i}^{(m)}.$$

Thus, EM only needs to be used to estimate the transition matrix entries. Let

$$\widetilde{\Psi}_N = \{r \in \Psi_N \ : \ r_{1,1} = 1, \ r_{N,N} = 1\},$$

denote the set of permutations of N elements with fixed endpoints. As in the general case, the E-step can be computed using summary statistics (for $t', t'' = 1, \ldots, N_m$)

$$
\begin{aligned}
\widetilde{\gamma}^{(m)} &= \sum_{r \in \widetilde{\Psi}_{N_m}} P[\mathbf{x}^{(m)} | \mathbf{r}, \widehat{\mathbf{A}}, \widehat{\boldsymbol{\pi}}] \\
\widetilde{\gamma}_{t',t''}^{(m)} &= \sum_{r \in \widetilde{\Psi}_{N_m}} P[\mathbf{x}^{(m)} | \mathbf{r}, \widehat{\mathbf{A}}, \widehat{\boldsymbol{\pi}}] \sum_{t=2}^{N_m} r_{t,t'} r_{t-1,t''},
\end{aligned}
$$

and setting $\bar{\alpha}_{t',t''}^{(m)} = \widetilde{\gamma}_{t',t''}^{(m)} / \widetilde{\gamma}$. The M-step (update for $\mathbf{A}^{k+1}$) remains unchanged.

3.6 Incorporating Prior Information

The EM algorithm can be easily modified to incorporate conjugate priors; these are Dirichlet priors for $\boldsymbol{\pi}$ and for each row of $\mathbf{A}$,

$$
P[\boldsymbol{\pi} | \mathbf{u}] \propto \prod_{i=1}^{|S|} \pi_i^{u_i - 1} \qquad \text{and} \qquad P[\mathbf{A} | \mathbf{v}] \propto \prod_{i=1}^{|S|} \prod_{j=1}^{|S|} A_{i,j}^{v_{i,j} - 1}, \tag{19}
$$

where the parameters u_i and $v_{i,j}$ should be non-negative in order to have proper priors [2]. The larger u_i is relative to the other $u_{i'}$, $i' \neq i$, the greater our prior belief that state i is an initial state rather than the others. Similarly, the larger $v_{i,j}$ relative to other $v_{i,j'}$ for $j' \neq j$, the more likely we expect, *a priori*, transitions from state i to state j relative to transitions from i to the other states.

Adding the logarithms of the priors in (19) to the complete log-likelihood (9), we find that incorporating priors into the EM algorithm only results in a change to the M-step. Consider the prior distribution on the initial state distribution; taking $u_i = c > 1$, for all i, has a *smoothing* effect, encouraging all of the states to have some mass in the initial state distribution. On the other hand, with $0 < c < 1$ will have a *shrinkage* effect, encouraging all of the mass to go to one (or a few) of the states. We can push even more aggressively for a sparse solution by choosing negative parameters for the Dirichlet distributions (which will become improper), as done in [7] for Gaussian

mixtures. When negative Dirichlet parameters are allowed, the M-step updates become

$$
\pi_i^{t+1} \;=\; \frac{\left(u_i + \sum_{m=1}^{T}\sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)}\right)_+}{\sum_{i=1}^{|S|}\left(u_i + \sum_{m=1}^{T}\sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)}\right)_+}
\tag{20}
$$

$$
A_{i,j}^{k+1} \;=\; \frac{\left(v_{i,j} + \sum_{m=1}^{T}\sum_{t'=1}^{N_m}\sum_{t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t-1,i}^{(m)} x_{t,j}^{(m)}\right)_+}{\sum_{j=1}^{|S|}\left(v_{i,j} + \sum_{m=1}^{T}\sum_{t'=1}^{N_m}\sum_{t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t-1,i}^{(m)} x_{t,j}^{(m)}\right)_+}.
\tag{21}
$$

where $(a)_+ = \max\{0, a\}$ is the so-called *positive part* operator.

4 Monte Carlo E-Step by Importance Sampling

For long sequences, the combinatorial nature of (15) and (16) (involving sums over all permutations of each sequence) may render exact computation impractical. In this section, we consider Monte Carlo approximate versions of the E-step, which avoid the combinatorial nature of its exact version. The Monte Carlo EM (MCEM) algorithm, based on an MC version of the E-step, was originally proposed in [26], and used ever since by many authors (recent work can be found in [3, 11] and references therein).

To lighten the notation in this section, we drop the superscripts from $(\mathbf{A}^k, \boldsymbol{\pi}^k)$, using simply $(\mathbf{A}, \boldsymbol{\pi})$ as the current parameter estimates. Moreover, we focus on a particular length-N path $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\} \subseteq S^N$ and drop the superscript (m); due to the independence of the paths, there's no loss of generality. Recall that $\mathbf{y}$ is a (shuffled) path, thus has no repeated elements.

The E-step (see (13) and (14)) consists of computing the conditional expectations $\bar{r}_{1,t'} = E\left[r_{1,t'} | \mathbf{x}, \mathbf{A}, \boldsymbol{\pi}\right]$ and $\bar{\alpha}_{t',t''} = E\left[\alpha_{t',t''} | \mathbf{x}, \mathbf{A}, \boldsymbol{\pi}\right]$. A naïve Monte Carlo approximation would be based on random permutations, sampled from the uniform distribution over Ψ_N. However, the reason to resort to approximation techniques in the first place is that Ψ_N is large, with only a small fraction of these random permutations having non-negligible posterior probability, $P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$; a very large

16

number of uniform samples is thus needed to obtain a good approximation to $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t',t''}$.

Ideally, we would sample permutations directly from the posterior $P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$; however, this would require determining its value for all $N!$ permutations. Instead, we employ *importance sampling* (IS) (see, *e.g.*, [15, 22], for an introduction to IS): we sample L permutations, $\mathbf{r}^1, \ldots, \mathbf{r}^L$, from a distribution $R[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$, from which it is easier to sample than $P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$, then apply a corrective re-weighting to obtain approximations to $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t',t''}$ (note that we are now using superscripts on $\mathbf{r}$ to index sample numbers, not to identify paths). The IS estimates are given by

$$
\bar{r}_{1,t'} \simeq \frac{\displaystyle\sum_{i=1}^{L} z_i\, r_{1,t'}^i}{\displaystyle\sum_{i=1}^{L} z_i}, \tag{22}
$$

$$
\bar{\alpha}_{t,t',t''} \simeq \frac{\displaystyle\sum_{i=1}^{L} z_i \sum_{t=2}^{N_m} r_{t,t'}^i\, r_{t-1,t''}^i}{\displaystyle\sum_{i=1}^{L} z_i}, \tag{23}
$$

where z_i, the correction factor (or weight) for sample $\mathbf{r}^i$, is given by

$$
z_i = \frac{P[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]}{R[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]}, \tag{24}
$$

the ratio between the desired distribution and the sampling distribution employed.

A relevant observation is that the target and sampling distributions only need to be known up to normalizing factors. Given $R'[\mathbf{r}] = Z_R\, R[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$ and $P'[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}] = Z_P\, P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$, for constants Z_R and Z_P, we can use

$$
z_i' = \frac{P'[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]}{R'[\mathbf{r}^i|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]} = \frac{Z_P}{Z_R}\, z_i, \tag{25}
$$

instead of z_i in (22) and (23); these sums will remain unchanged since the factor Z_P/Z_R will appear both in the numerator and denominator, thus cancelling out.

The remainder of this section contains the description of an IS scheme, including the derivation

of closed form expressions for both the sampling distribution, R, and the sample weights, z_i. We conclude the section by mentioning other sampling variants and presenting experimental results.

4.1   Sampling Scheme

Let $\mathbf{f} = \{f_1, \ldots, f_{|S|}\} \in \{0,1\}^{|S|}$ be a sequence of binary flags. Given some probability distribution $\mathbf{p} = \{p_1, p_2, \ldots, p_{|S|}\}$ on the set of states, S, denote by $\mathbf{p} \cdot \mathbf{f}$ the restriction of $\mathbf{p}$ to those elements of S that have corresponding flag f_i set to 1, that is,

$$(\mathbf{p} \cdot \mathbf{f})_i = \frac{p_i \, f_i}{\sum\limits_{j=1}^{|S|} p_j \, f_j}, \quad \text{for } i = 1, 2, \ldots, |S|. \tag{26}$$

The proposed sampling scheme is defined as follows:

Step 1: Let $\mathbf{f} = \{f_1, \ldots, f_{|S|}\}$ be initialized according to $f_i = \mathbb{I}_{\{i \in \mathbf{y}\}}$, where $\mathbb{I}_{\{\}}$ denotes the indicator function.

Obtain one sample from S according to the distribution $\boldsymbol{\pi} \cdot \mathbf{f}$. Let the obtained sample be denoted s; of course, one and only one element of $\mathbf{y}$ is equal to s.

Locate the position t of s in $\mathbf{y}$; that is, find t such that $y_t = s$. Set $\tau_1 = t$.

Set $f_s = 0$ (preventing y_t from being sampled again). Set $i = 2$.

Step 2: Let $\mathbf{p} = \{A_{s,1}, \ldots, A_{s,|S|}\}$ be the sth row of the transition matrix.

Obtain a sample s' from S, according to the distribution $\mathbf{p} \cdot \mathbf{f}$.

Find t such that $y_t = s'$. Set $\tau_i = t$. Set $f_{s'} = 0$.

Step 3: If $i < N$, then set $s \leftarrow s'$, set $i \leftarrow i + 1$, go back to Step 2; otherwise, stop.

4.1.1 Sampling Distribution

Before deriving the form of the distribution R, let us begin by writing the target distribution $P[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]$ explicitly. Using Bayes law,

$$P[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] = \frac{P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}]P[\boldsymbol{\tau}]}{P[\mathbf{y}|\mathbf{A}, \boldsymbol{\pi}]}, \tag{27}$$

since $\boldsymbol{\tau}$ does not depend *a priori* on $\mathbf{A}$ or $\boldsymbol{\pi}$. Based on our assumption that all permutations are equiprobable we have $P[\boldsymbol{\tau}] = \mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}}/N!$. Noticing that the denominator in (27) is just a normalizing constant independent of $\boldsymbol{\tau}$, we have

$$P[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}] \propto \mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}} \, P[\mathbf{y}|\boldsymbol{\tau}, \mathbf{A}, \boldsymbol{\pi}] = \mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}} \left(\pi_{y_{\tau_1}} \prod_{t=2}^{N} A_{y_{\tau_{t-1}}, y_{\tau_t}} \right). \tag{28}$$

For the sake of notational economy, we will write simply $R[\boldsymbol{\tau}]$ to represent $R[\boldsymbol{\tau}|\mathbf{y}, \mathbf{A}, \boldsymbol{\pi}]$. The sequential nature of the sampling scheme suggests a factorization of the form

$$R[\boldsymbol{\tau}] = R[\tau_1] \, R[\tau_2|\tau_1] \, R[\tau_3|\tau_2, \tau_1] \, \cdots \, R[\tau_N|\tau_{N-1}, \ldots, \tau_1]. \tag{29}$$

For Step 1 of the sampling scheme, it's clear that, for $\tau_1 = 1, \ldots, N$,

$$R[\tau_1] \propto \pi_{y_{\tau_1}}. \tag{30}$$

For the i-th iteration, we have,

$$R[\tau_i|\tau_{i-1}, \ldots, \tau_1] = A_{y_{\tau_{i-1}}, y_{\tau_i}} \, \phi_i(\tau_{i-1}, \ldots, \tau_1) \, \mathbb{I}_{\{\tau_i \notin \{\tau_{i-1}, \ldots, \tau_1\}\}}, \tag{31}$$

with

$$\phi_i(\tau_{i-1}, \ldots, \tau_1) = \left(\sum_{t \notin \{\tau_{i-1}, \ldots, \tau_1\}} A_{y_{\tau_{i-1}}, y_t} \right)^{-1}.$$

Inserting (30) and (31) into (29), we finally have

$$R[\boldsymbol{\tau}] \propto \left[\pi_{y_{\tau_1}} \prod_{i=2}^{N} A_{y_{\tau_{i-1}}, y_{\tau_i}}\right] \left[\prod_{i=2}^{N} \phi_i(\tau_{i-1}, \dots, \tau_1)\right] \left[\prod_{i=2}^{N} \mathbb{I}_{\{\tau_i \notin \{\tau_{i-1}, \dots, \tau_1\}\}}\right] ; \tag{32}$$

observe that the third factor in the r.h.s. of (32) is simply the indicator that $\boldsymbol{\tau}$ is a permutation, i.e., is equal to $\mathbb{I}_{\{\boldsymbol{\tau} \in \Psi_N\}}$, for any $\boldsymbol{\tau} \in \{1, \dots, N\}^N$.

Dividing (28) by (32) we obtain the correction factor z for a permutation sample $\boldsymbol{\tau}$ generated using this sequential scheme as

$$z = \left(\prod_{i=2}^{N} \phi_i(\tau_{i-1}, \dots, \tau_1)\right)^{-1} = \prod_{i=2}^{N} \sum_{t \notin \{\tau_{i-1}, \dots, \tau_1\}} A_{y_{\tau_{i-1}}, y_t}. \tag{33}$$

With this quantity in hand, we have all the ingredients needed to produce IS estimates of $\bar{r}_{1,t'}$ and $\bar{\alpha}_{t,t',t''}$. Notice that computing the terms ϕ_i, thus computing z, is easy since these factors are the normalization terms for the distributions $\mathbf{p} \cdot \mathbf{f}$, which are already computed while performing each iteration of Step 2. Thus, we just need to store the product of these normalizing constants to finally obtain the weight z.

4.1.2 Known Endpoints

In the case where the endpoints are known, we fix $\tau_1 = 1$, $\tau_N = N$, and set $f_1 = 0$ and $f_N = 0$ in Step 1; the remainder of the procedure is unchanged. Based on these constraints, the importance sampling weight takes a slightly different form:

$$z = \pi_{y_1} A_{y_{\tau_{N-1}}, y_N} \prod_{i=2}^{N-1} \sum_{t \notin \{\tau_{i-1}, \dots, \tau_1\}} A_{y_{\tau_{i-1}}, y_t}. \tag{34}$$

4.2 Hierarchical Sampling Schemes

In addition to the sampling scheme that we have just described, we have also developed other sampling schemes that work in a hierarchical, rather than sequential, fashion. For the sake of space, we refrain from describing these other sampling schemes; detailed descriptions can be found in [20].

In particular, we have developed a two-stage hierarchical scheme and a fully hierarchical scheme. In the two-stage method, the first stage samples from the collection of all possible transitions occurring in a path; then the second stage samples from the distribution on all arrangements of these transitions, to form a permutation. In the fully hierarchical method, the first stage samples a suitable set of transitions, say $\mathcal{G}_1$; then, the following stage samples a suitable collection of pairs of elements of $\mathcal{G}_1$, yielding a collection of quadruples, $\mathcal{G}_2$, and the procedure is repeated until a permutation is obtained.

4.3   Performance Assessment

A standard error metric for comparing two distributions P and $\widehat{P}$ taking values on the finite set Ψ_N is the ℓ_1 distance, defined as

$$\|P - \widehat{P}\|_1 = \sum_{\mathbf{r} \in \Psi_N} \left| P[\mathbf{r}] - \widehat{P}[\mathbf{r}] \right|. \tag{35}$$

Given a set of permutations, $\{\mathbf{r}^1, \ldots, \mathbf{r}^L\}$, drawn from the sampling distribution R along with the corresponding weights, $z_1, \ldots, z_L$, we can compute the empirical distribution $\widehat{P}_R$ induced on Ψ_N according to

$$\widehat{P}_R[\mathbf{r}] = \left(\sum_{i=1}^{L} z_i \right)^{-1} \sum_{i=1}^{L} z_i \, \mathbb{I}_{\{\mathbf{r}^i = \mathbf{r}\}}. \tag{36}$$

Notice that the Monte Carlo sufficient statistics $\widehat{\alpha}_{t',t''}^{(m)}$ and $\widehat{r}_{1,t'}^{(m)}$ are just sums of terms $\widehat{P}_R[\mathbf{r}]$, $e.g.$, $\widehat{\alpha}_{t',t''} = \sum_{\mathbf{r} \in \Psi_N} \widehat{P}_R[\mathbf{r}] \sum_{t=2}^{N_m} r_{t-1,t''} r_{t,t'}$. Thus,

$$\left| \bar{\alpha}_{t',t''}^{(m)} - \widehat{\alpha}_{t',t''}^{(m)} \right| \le \left\| P - \widehat{P}_R \right\|_1 \qquad \text{and} \qquad \left| \bar{r}_{1,t'}^{(m)} - \widehat{r}_{1,t'}^{(m)} \right| \le \left\| P - \widehat{P}_R \right\|_1,$$

showing that if the ℓ_1 error between the true distribution on permutations and the empirical importance sampling distribution is small, then all of the estimated sufficient statistics will be close to the corresponding exact value.

We have evaluated the performance of various sampling schemes via simulation, considering three scenarios concerning the distribution over all permutations: 1) roughly uniform, 2) moderately

peaked, and 3) highly concentrated around just a few permutations. The scenario 1) is typical of the first EM iterations; scenario 2) is typical during intermediate EM iterations; the third scenario is typical when EM is near convergence. We consider a length-8 path with known endpoints, thus with $6! = 720$ possible orderings. This length is long enough to suggest how each sampling scheme will perform for longer paths, while still allowing enumeration of all orderings.

Figure 1 depicts the ℓ_1 error between the true and IS-based distributions on permutations, as a function of the number of samples. The curve labelled "True Dist" corresponds to sampling from the true distribution, shown as a reference, is only possible when we can enumerate all permutations. The "Causal IS" curve corresponds to the scheme described in Section 4.1. The "Two Stage" and "Hierarchical" curves depict the performance for the hierarchical schemes mentioned in Section 4.2. Finally, "Random" refers to an approach where we sample from a uniform distribution on permutations, shown as a baseline comparison. Each curve was generated by averaging over 50 Monte Carlo simulations. These curves depict performance using up to 500 samples for a path with 720 possible orderings, which is actually quite a generous helping of samples. In experiments with Internet data we have encountered paths of up to length 27, and observed good reconstruction performance using as few as 2000 samples (notice that $27! \simeq 10^{28}$). Thus, performance for very few samples is of great interest. As expected, all of the sampling schemes give the same performance when distribution is roughly uniform. However, as the distribution becomes more concentrated, there is a clear difference between the various sampling schemes. The uniform sampling scheme naturally performs the worst on more concentrated distributions. Of the schemes that are practical for long paths, these results indicate that the causal sampling scheme performs the best.

In terms of computational complexity, the causal sampler is the simplest and fastest, requiring only $O(N)$ operations per sample (N is the path length). The two-stage sampler requires $O\big((N/2)!\big)$ operations per sample, while the fully hierarchical has computational complexity $O(N^2 \log N)$ operations per sample. The conclusion is that the causal sampling method described above is simple to implement, fast, and it empirically outperforms more computationally complex schemes.

To compare the efficacy of each sampling scheme within the EM algorithm, we generated a random network with 250 nodes and simulated 60 random sample paths through this network,
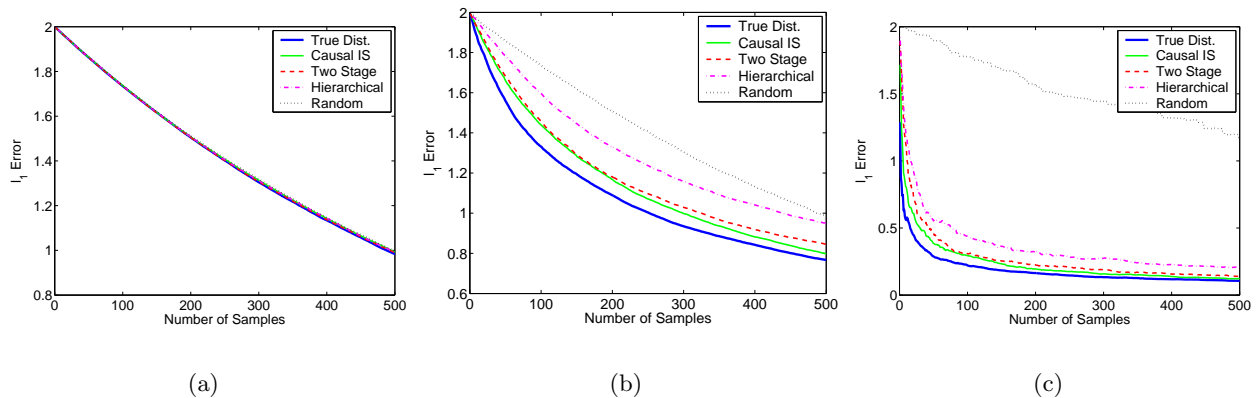
22

Figure 1: ℓ_1 error as a function of the number of importance samples drawn for various sampling schemes in the following scenarios: (a) a roughly uniform distribution on the permutations, (b) moderately peaked distribution, (c) highly concentrated distribution. The curves in each figure were calculated by averaging over 50 Monte Carlo simulations.

ranging in length from 4 to 10 hops. Then, we estimated a probability transition matrix for the network using the EM algorithm with different IS-based E-steps (with known endpoints for each path). Figure 2 depicts the marginal log-likelihood of the data, computed according to (4) using the probability transition matrices returned by the EM algorithm, for a number of samples-per-path between 20 and 100. The horizontal dashed line at the top marks the marginal log likelihood computed using a transition matrix estimated from correctly ordered paths.

5   Monotonicity and Convergence

Well-known convergence results due to Wu and Boyles [4, 27] guarantee convergence of our EM algorithm when the exact E-step is used. Let $\boldsymbol{\theta}^k = \left(\mathbf{A}^k, \boldsymbol{\pi}^k\right)$ denote parameter estimates calculated at the kth EM iteration using the exact EM expressions. By choosing $\boldsymbol{\theta}^{k+1} = \left(\mathbf{A}^{k+1}, \boldsymbol{\pi}^{k+1}\right)$ according to (11) in the M-step, our iterates satisfy the *monotonicity property*:

$$Q\left(\boldsymbol{\theta}^{k+1}; \boldsymbol{\theta}^k\right) \; \geq \; Q\left(\boldsymbol{\theta}^k; \boldsymbol{\theta}^k\right). \tag{37}$$

The marginal log-likelihood (4) is continuous in its parameters $\mathbf{A}$ and $\boldsymbol{\pi}$ and it is bounded above. In this setting the monotonicity property (37) guarantees that each exact EM update monotonically
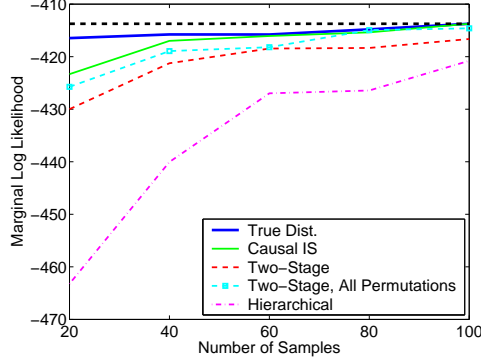
Figure 2: Using various approximate E-steps in the EM algorithm for estimating the Markov transition matrix of a simulated network. The horizontal dashed line at the top of the figure marks the marginal log likelihood of the data using the transition matrix derived using the correctly ordered paths. The curves in this figure correspond to the average over 10 Monte Carlo simulations.

increases the marginal log-likelihood, so the EM iterates converge to a local maximum.

When Monte Carlo methods are used in the E-step monotonicity is no longer guaranteed since the M-step solves

$$\widehat{\boldsymbol{\theta}}^{k+1} \equiv \left(\widehat{\mathbf{A}}^{k+1}, \widehat{\boldsymbol{\pi}}^{k+1}\right) \quad = \quad \arg\max_{\mathbf{A},\boldsymbol{\pi}} \widehat{Q}\left(\mathbf{A}, \boldsymbol{\pi}; \mathbf{A}^k, \boldsymbol{\pi}^k\right),$$

where $\widehat{Q}$ is defined analogously to Q but with terms $\bar{\alpha}_{t',t''}^{(m)}$ and $\bar{r}_{1,t'}^{(m)}$ replaced by $\widehat{\alpha}_{t',t''}^{(m)}$ and $\widehat{r}_{1,t'}^{(m)}$, their corresponding importance sampling approximations. Consequently, care must be taken to ensure that $\widehat{Q}$ approximates Q well enough so that the EM algorithm is not swamped with error from the Monte Carlo estimates.

To illustrate this issue, consider the following synthetic example. We generate 40 co-occurrence observations by taking a random walk on a graph with 140 vertices. Each co-occurrence has between 4 and 8 vertices. Figure 3(a) plots $Q(\boldsymbol{\theta}^k; \boldsymbol{\theta}^{k-1})$ for the exact E-step, along with $\widehat{Q}(\widehat{\boldsymbol{\theta}}^{k+1}; \widehat{\boldsymbol{\theta}}^k)$ and $Q(\widehat{\boldsymbol{\theta}}^{k+1}; \widehat{\boldsymbol{\theta}}^k)$ for the Monte Carlo EM algorithm using only 10 importance samples per co-occurrence. Although $\widehat{Q}(\widehat{\boldsymbol{\theta}}^{k+1}; \widehat{\boldsymbol{\theta}}^k)$ increases at each iteration, $Q(\widehat{\boldsymbol{\theta}}^{k+1}; \widehat{\boldsymbol{\theta}}^k)$ clearly does not and the monotonicity property does not hold. This is apparent in Figure 3(b), where the dash-dot line shows the progress of the marginal log-likelihood (our optimization criterion) for the 10 sample Monte Carlo EM algorithm. When enough importance samples are used the Monte Carlo EM algorithm performs
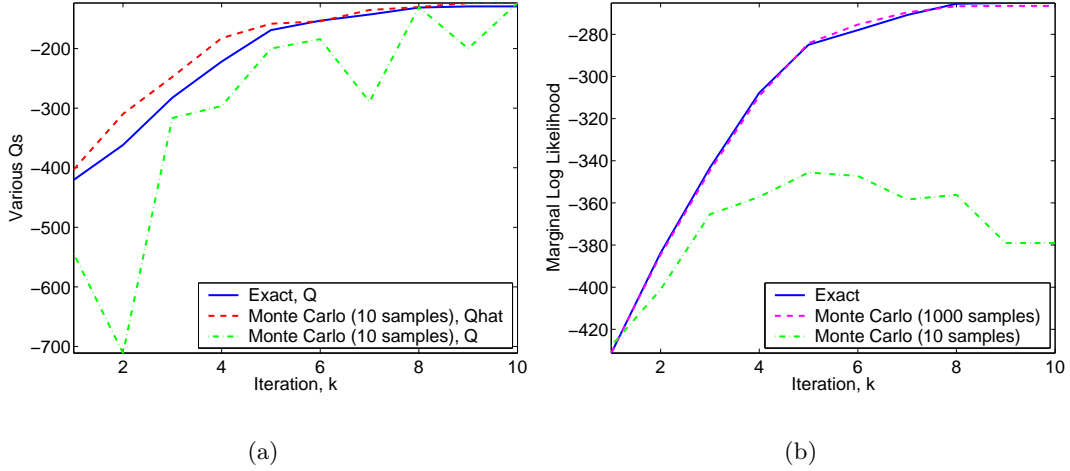
Figure 3: An example with simulated observations illustrating that the Monte Carlo EM algorithm may not result in monotonic increase of the marginal log-likelihood if too few Monte Carlo samples are used. The solid line in (a) is $Q(\boldsymbol{\theta}^{k+1};\boldsymbol{\theta}^k)$ for exact EM iterations, the dashed line is $\widehat{Q}(\widehat{\boldsymbol{\theta}}^{k+1};\widehat{\boldsymbol{\theta}}^k)$ and the dash-dot line is $Q(\widehat{\boldsymbol{\theta}}^{k+1};\widehat{\boldsymbol{\theta}}^k)$ for Monte Carlo EM iterations using only 10 samples. Even though $\widehat{Q}$ increases monotonically, Q may not be monotonic for the Monte Carlo EM algorithm. Figure (b) depicts the marginal log-likelihood for exact EM iterates and for two versions of the Monte Carlo EM. Monte Carlo EM performance closely resembles that of the exact EM algorithm when sufficiently many importance samples are used.

comparably to the exact EM algorithm; see the dashed line in Figure 3(b) corresponding to a Monte Carlo EM algorithm using 1000 importance samples per co-occurrence. All three instances of the EM algorithm used in this example start from the same initialization.

Recently, researchers have considered the question of how many importance samples should be used in a Monte Carlo E-step [3, 5, 11]. The goal is to balance monotonicity and computational efficiency by using enough samples to have a good chance at monotonicity while not using excessively many samples. Booth et al. [3] argue that if the same number of importance samples is used at each EM iteration, then the algorithm will eventually be swamped by Monte Carlo error and will not converge. They also suggest requiring that a convergence criterion be satisfied on multiple successive iterations since the criterion may be met prematurely due to poor Monte Carlo approximations.

Caffo et al. [5] propose a method for automatically adapting the number of Monte Carlo samples used at each EM iteration. To lighten notation, we drop the superscripts k and $k+1$. Let $\Delta(\boldsymbol{\theta}) = Q(\boldsymbol{\theta};\boldsymbol{\theta}') - Q(\boldsymbol{\theta}';\boldsymbol{\theta}')$ and $\widehat{\Delta}(\boldsymbol{\theta}) = \widehat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}') - \widehat{Q}(\boldsymbol{\theta}';\boldsymbol{\theta}')$. Furthermore, let $\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \widehat{Q}(\boldsymbol{\theta};\boldsymbol{\theta}')$, where

$\boldsymbol{\theta}' = \boldsymbol{\theta}^k$ was determined at the previous EM iteration. Recall that L importance samples are used to calculate $\widehat{Q}$. The algorithm of Caffo et al. is based on a Central Limit Theorem-like approximation in which they show that $\sqrt{L}\big(\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})\big)$ converges in distribution to the standard normal. Observe that the monotonicity property (37) is equivalent to the condition $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$. Although $\Delta(\widehat{\boldsymbol{\theta}})$ cannot be computed without computing the exact sufficient statistics $\{\bar{\alpha}_{t',t''}^{(m)}\}$ and $\{\bar{r}_{1,t'}^{(m)}\}$, we can compute $\widehat{\Delta}(\widehat{\boldsymbol{\theta}})$. Their scheme then amounts to increasing the number of Monte Carlo samples until $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) > \epsilon$ for a user-specified $\epsilon > 0$. Then, applying an asymptotic standard normal tail approximation, they obtain a statement of the form $\Pr\big(\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) \geq \epsilon\big) \leq \delta(\epsilon)$. Based on this statement they claim that monotonicity holds with probability at least $1 - \delta(\epsilon)$. They further remark that if a different ϵ_k is chosen at each iteration, so that $\sum_{k=1}^{\infty} \delta(\epsilon_k) < \infty$, then, by the Borel-Cantelli Lemma, $\Pr\big(\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) \geq \epsilon_k \text{ i.o.}\big) = 0$, so there exists a $K > 0$ such that $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) < \epsilon_k$ for all $k \geq K$ with probability 1; *i.e.*, eventually every EM update is monotonic. Of course, in practice, the algorithm is terminated after a finite number of iterations, so we may never reach the stage where all iterates are monotonic.

Notice that for the monotonicity condition $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$ to truly hold in the above framework, the events

$$\left\{\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) \leq \epsilon\right\} \quad \text{and} \quad \left\{\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) \geq \epsilon\right\}$$

must occur simultaneously. Because the probabilistic bound above only addresses one of these events we refer to this type of result as guaranteeing an (ϵ, δ)-*probably approximately monotonic* update, or PAM for short. More generally, an (ϵ, δ)-PAM result states that with probability at least $1 - \delta$, the update will be ϵ-approximately monotonic; *i.e.*, $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) \leq \epsilon$ implies $\Delta(\widehat{\boldsymbol{\theta}}) \geq -\epsilon$, because, by definition, $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) \geq 0$.

Rather than resorting to asymptotic approximations to obtain such a result, we can take advantage of the specific form of Q in our problem to obtain the finite-sample PAM result next presented. Recall that independent importance samples are drawn for each co-occurrence observation in the Monte Carlo E-step. Denote by L_m the number of importance samples used to compute sufficient statistics for observation $\mathbf{x}^{(m)}$. Exact E-step computation for this observation requires $N_m!$ opera-

tions. Similarly, we should expect that larger observations will require more importance samples for two reasons: 1) there are more sufficient statistics associated with this observation (N_m^2 in total), and 2) there are more ways to shuffle these observations.

In the previous section we derived closed form expressions for the importance sample weights, $z_i = P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]/R[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$, where P is the target distribution and R is the importance sampling distribution. A key assumption was made that P is absolutely continuous with respect to R; that is, $P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}] = 0$ for every permutation $\mathbf{r}$ with $R[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}] = 0$. We adopt the convention $0/0 = 0$ so that $z_i = 0$ for such samples. This guarantees that $z_i < \infty$. The bounds below depend on the quality of our importance sample estimators as gauged by

$$b_m = \max_{\mathbf{r} \in \Psi_{N_m}} \frac{P[\mathbf{r}|\mathbf{x}^{(m)}, \mathbf{A}, \boldsymbol{\pi}]}{R[\mathbf{r}|\mathbf{x}^{(m)}, \mathbf{A}, \boldsymbol{\pi}]}. \tag{38}$$

Because the set Ψ_{N_m} is finite, P and R have finite support, and the maximum is well-defined. If R matches the target distribution P well then b_m should not be very large.

There is one other subtlety we will account for in our bounds. Because $\widehat{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}')$ has terms $\log A_{i,j}$ and $\log \pi_i$, in practice we typically bound $A_{i,j}$ and π_i away from zero to ensure that $\widehat{Q}$ does not go to $-\infty$. To this end, we will assume that $\widehat{A}_{i,j} \geq \theta_{\min}$ and $\widehat{\pi}_i \geq \theta_{\min}$ for some $0 < \theta_{\min} < |S|^{-1}$. The upper bound on $\theta_{\min}$ ensures it is still possible to satisfy the constraints (1). Generally we choose $\theta_{\min}$ very close to zero; at machine precision, for example.

We have the following finite-sample PAM result for our Monte Carlo EM algorithm.

Theorem 1. *Let $\epsilon > 0$ and $\delta > 0$ be given and assume there exists $\theta_{\min} \in (0, |S|^{-1})$ such that $A'_{i,j} \geq \theta_{\min}$ and $\pi'_i \geq \theta_{\min}$ for all i and j. If*

$$L_m = \frac{2T^2 N_m^4 b_m^2 |\log \theta_{\min}|^2}{\epsilon^2} \log\left(\frac{2N_m^2}{1 - (1 - \delta)^{1/T}}\right) \tag{39}$$

importance samples are used for the mth observation then $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) < \epsilon$ with probability greater than $1 - \delta$.

The proof of Theorem 1 appears in Appendix A. Because $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) \geq 0$ by definition, the theorem guarantees that $\Delta(\widehat{\boldsymbol{\theta}}) > -\epsilon$ with probability greater than $1 - \delta$.

Recall that exact E-step computation requires $N_m!$ operations for the mth observation. The bound above stipulates that the number of importance samples required is proportional to $N_m^4 \log N_m^2$, and generating one importance sample requires N_m operations. Thus, the computational complexity of a PAM Monte Carlo update only depends on $N_m^5 \log N_m^2$, which clearly demonstrates that the computational complexity of the Monte Carlo E-step depends polynomially on the N_m in comparison to exponential dependence for the exact E-step.

To put this result in perspective, observe that the value of L_m given by (39) is roughly a factor of T away from the value we would expect based on an asymptotic variance calculation. Ignoring constants and log terms, for fixed $\boldsymbol{\theta}$ we have

$$
\begin{aligned}
\text{Var}\left(\widehat{\Delta}(\boldsymbol{\theta})\right) &\approx \text{Var}\left(\sum_{m=1}^{T}\sum_{t',t''=1}^{N_m}\widehat{\alpha}_{t',t''}^{(m)} + \sum_{m=1}^{T}\sum_{t'=1}^{N_m}\widehat{r}_{1,t'}^{(m)}\right) \\
&= \sum_{m=1}^{T}\text{Var}\left(\sum_{t',t''=1}^{N_m}\widehat{\alpha}_{t',t''}^{(m)} + \sum_{t'=1}^{N_m}\widehat{r}_{1,t'}^{(m)}\right),
\end{aligned}
$$

since independent sets of importance samples are used to calculate sufficient statistics for different observations. It is easily shown that the variance of an individual approximate statistic $\widehat{\alpha}_{t',t''}^{(m)}$ or $\widehat{r}_{1,t'}^{(m)}$ decays according to the parametric rate; *i.e.*, $\text{Var}(\widehat{\alpha}_{t',t''}^{(m)}) \simeq 1/L_m$. In total, there are N_m^2 sufficient statistics for the mth observation, and they are all potentially correlated since they are functions of the same set of importance samples. Then we have

$$
\text{Var}\left(\widehat{\Delta}(\boldsymbol{\theta})\right) \approx \sum_{m=1}^{T}\frac{\left(N_m^2\right)^2}{L_m}.
$$

To drive $\text{Var}\left(\widehat{\Delta}(\boldsymbol{\theta})\right)$ down to a constant level, independent of T and N_m, we need $L_m \propto TN_m^4$. The additional factor of T in our bound is essentially an artifact from the union bound.

Although the PAM result is pleasing, we would prefer to guarantee *monotonicity* with high probability, not just *approximate* monotonicity. Let $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta};\boldsymbol{\theta}')$. By bounding $\Delta(\widehat{\boldsymbol{\theta}}) - \Delta(\boldsymbol{\theta}^*)$ instead of $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}})$ we obtain the following stronger result guaranteeing monotonicity with high probability. Instead of restricting $A_{i,j} \geq \theta_{\min}$ and $\pi_i \geq \theta_{\min}$, we need to assume the variables

$\bar{\alpha}_{t',t''}^{(m)}$ and $\bar{r}_{1,t'}^{(m)}$ are bounded away from zero. This is stronger than the previous assumption in the sense that it implies $A_{i,j}$ and π_i are bounded away from zero.

Theorem 2. *Let $\delta > 0$ be given and assume there exists $\lambda > 0$ such that $\bar{\alpha}_{t',t''}^{(m)} > \lambda$ and $\bar{r}_{1,t'}^{(m)} > \lambda$, for all t' and t''. If*

$$L_m \;=\; \frac{27 b_m}{\lambda} \left(\frac{2 \sum_{m=1}^{T} N_m + \Delta(\boldsymbol{\theta}^*)}{\Delta(\boldsymbol{\theta}^*)} \right)^2 \log \left(\frac{4 \sum_{m=1}^{T} N_m^2}{\delta} \right) \tag{40}$$

importance samples are used for the mth observation, then $\Delta(\widehat{\boldsymbol{\theta}}) \geq 0$ with probability at least $1 - \delta$.

The proof of Theorem 2 appears in Appendix B. Similar to the PAM bound given in Theorem 2, the computational complexity required for a *probably monotonic* update also depends polynomially on N_m.

Note that L_m depends on $\Delta(\boldsymbol{\theta}^*)$. By definition, $\Delta(\boldsymbol{\theta}^*) \geq 0$ at every iteration, and typically $\Delta(\boldsymbol{\theta}^*)$ is large at earlier EM iterations and approaches zero as the algorithm converges. This dependence supports the observation of Booth et al. mentioned earlier, that the number of importance samples ought to increase at each iteration.

The main assumption of Theorem 2 is that the sufficient statistics are bounded away from zero at each iteration. We motivate this assumption by observing that if the algorithm is initialized with $\pi_i^0 > 0$ and $A_{i,j}^0 > 0$ for all i, j then, recalling the closed form E-step expressions, the sufficient statistics do not vanish in a finite number of EM iterations.

Note that if we use different δ_k at each EM iteration, chosen such that $\sum_{k=1}^{\infty} \delta_k < \infty$, then by the Borel-Cantelli Lemma one can argue that $\Pr\left(\Delta(\widehat{\boldsymbol{\theta}}) < 0 \text{ i.o.}\right) = 0$. In other words, eventually all EM iterates result in a monotonic increase of the marginal log-likelihood.

In addition to demonstrating that the Monte Carlo EM algorithm has polynomial computational complexity, these bounds give a useful guideline for determining how many importance samples should be used. However, because they involve worst-case analysis, the numbers of samples dictated by these bounds tend to be on the conservative side. For example, in the Internet experiments described in Section 6, $T = 249$ and $\bar{N} = 17$. Theorem 2 suggests that roughly 72 million importance samples should be used per observation. However, in our experiments we find

that the algorithm exhibits reasonable performance on this data set using as few as $2,000$ samples per observation. Of course, in practice, we do not have direct access to the parameters b_m, λ, or $\Delta(\boldsymbol{\theta}^*)$, so these bounds cannot be used as explicit guidelines.

6   Experimental Results

In this section, we evaluate the performance of our *network inference from co-occurrences* (NICO) algorithm on simulated data and on data gathered from the public Internet. In the results reported below, network reconstructions are obtained by first estimating an initial state distribution and probability transition matrix via the EM algorithm. Then, we compute the most likely order of each observation according to the inferred model and use this ordering to reconstruct a feasible network. The EM algorithm cannot be guaranteed to converge to a global maximum (the marginal log-likelihood is not concave) and there may even be multiple global maxima. To address this issue, we rerun the EM algorithm from multiple random initializations and report the collective results.

We compare the performance of our algorithm with that of the *frequency method* (FM), defined in [21] and mentioned in Section 1. The FM also reconstructs a network topology by estimating an order of the vertices in each observation. This method individually determines each path ordering independently by sorting the elements in the path according to a score computed from pair-wise co-occurrence frequencies involving the source and destination of the path. It is possible that multiple vertices may receive identical FM scores, in which case their sorting would be arbitrary (one could exchange elements with identical scores without violating the FM criteria). In fact, we observe this phenomenon in many of our experiments. Ties are resolved by choosing a random order for elements with identical scores. Multiple restarts are also performed using the FM, yielding a collection of feasible solutions.

The quality of a network reconstruction is determined by a quantity we term the *edge symmetric difference* error. Because the nodes in the network have unique labels, the goal of any reconstruction scheme is to determine which vertices are connected by an edge. The edge symmetric difference error is defined as the sum of the number of false positives (edges appearing in the reconstructed network which do not exist in the true network) and the number of false negatives (edges in the

30

true network not appearing in the reconstructed network).

6.1   Simulated Networks

Our synthetic data is obtained as described next. A network is generated according to a random geometric graph model: 50 vertices are thrown at random in the unit square, and two vertices are connected with an edge if the Euclidean distance between them is less than or equal to $\sqrt{\log(50)/50}$. This threshold guarantees that the graph is connected with high probability. Groups of nodes are randomly chosen as sources and destinations, transmission paths are generated between each source-destination pair according to either a shortest path or random routing model, and then co-occurrence observations are formed from each path. We keep the number of sources fixed at 5 and vary the number of destinations between 5 and 40, to see how the number of observations effects performance. Each experiment is repeated on 100 different topologies, using 10 restarts of both NICO and the FM per configuration. Exact E-step calculation is used for observations with $N_m \leq 12$, and causal importance sampling (2000 samples) is used for longer paths. The longest observation in our data was obtained by random routing and has $N_m = 19$ (notice that $19! \simeq 10^{17}$).

Figure 4 plots edge symmetric difference performance for synthetic data generated using (a) shortest path routing and (b) random routing. The edge symmetric difference error is computed between the inferred network and the graph obtained from correctly ordered observations. Of the 10 solutions corresponding to different NICO initializations, we use the one based on parameter estimates yielding the highest likelihood score. For this simulation, the most likely NICO solution also always resulted in the best edge symmetric difference error.

The FM does not provide a similar mechanism for ranking different solutions. A possible heuristic would be to choose the sparsest solution (with fewest edges). The figures show performance for both this heuristic, and clairvoyantly choosing the best (lowest error) solution of the 10. In fact, using the sparsest solution does better than just choosing a FM solution at random but not as well as using the clairvoyant best. In these simulations, NICO consistently outperform the FM.

Notice that both algorithms exhibit their worst performance at an intermediate number of destinations. When very few destinations are used the measured topology closely resembles a tree,

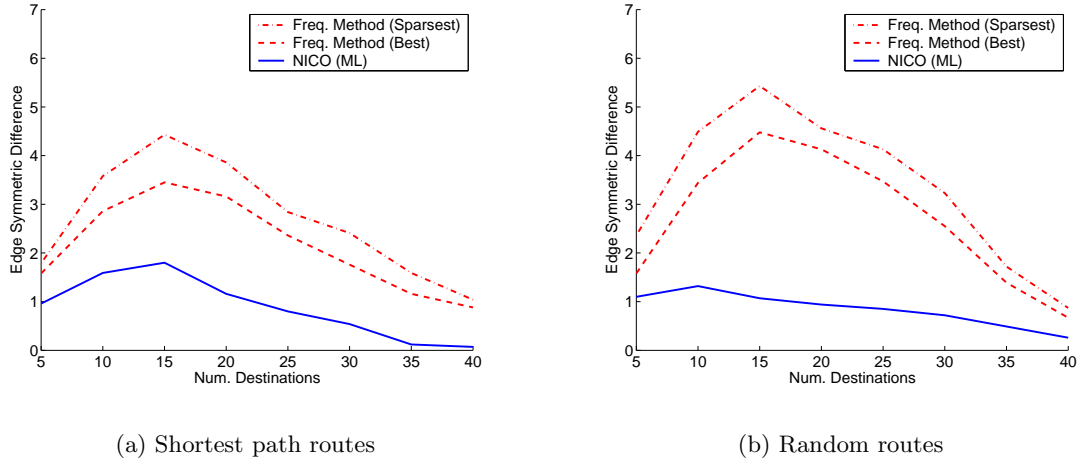| (a) Shortest path routes | (b) Random routes |

Figure 4: Edge symmetric differences between inferred networks and the network one would obtain using co-occurrence measurements arranged in the correct order. Performance is averaged over 100 different network configurations. For each configuration 10 NICO and FM solutions are obtained via different initializations. We then choose the NICO solution yielding the largest likelihood, and compare with both the sparsest and clairvoyant best FM solution.

regardless of the underlying routing mechanism. Relative frequencies of co-occurrence accurately reflect the network distance of each internal vertex from the path endpoints. At the other extreme, when many destinations are used, there is significant overlap among the co-occurrence observations which aids in localizing vertices. In general, the FM seems to be much more sensitive to the amount of data available.

As expected, the FM generally performs better on shortest path data than it does on random routes. When routes are generated randomly the corresponding topology is less tree-like and pairwise co-occurrence frequencies do not reflect network distances. Because NICO is not based on a particular routing paradigm it performs similarly in both scenarios, possibly even a little better when routing is random.

6.2 Internet Data

We have also studied the performance of our algorithm on co-occurrence observations gathered from the Internet. Using `traceroute` we have collected data describing roughly 250 router-level paths from sources at the University of Wisconsin-Madison, the *Instituto Superior Técnico* in Lisbon, and

Rice University to 83 servers affiliated with corporations, universities, and governments around the world. Our motivation for using this type of data is two-fold. First, `traceroute` allows us to measure the true order of elements in each path so that we have a ground truth to validate our results against. Second, and more importantly, the data comes from a real network where, presumably, paths are not generated according to a first-order Markov model. This allows us to gauge the robustness of the proposed model and to evaluate how well it generalizes to realistic scenarios. The ground truth network contains a total of 1105 nodes and 1317 edges, and the longest observed path has length 27.

For this data set we rerun FM and NICO each from 50 random initializations and look at performance across all solutions rather than focusing on the maximum likelihood or clairvoyant best. The exact E-step is used to compute $\bar{\alpha}$ for paths of up to 9 hops. For paths longer than 9 hops, we use the causal importance sampling described in Section 4.1, with 2000 samples per observation.

Minimum, median, and maximum edge symmetric difference errors are shown in Figure 5. Both algorithms have seemingly high error rates, as there are roughly 1300 links in the true network. However keep in mind that both algorithms are attempting to fill in the entries of a roughly 1100×1100 matrix. For 50 networks constructed by choosing a random order for the elements of each observation the average edge symmetric difference error was 4300, so both algorithms are indeed doing considerably better than random guessing. NICO performance is again noticeably better than that of the FM; the NICO average error is better than that of the best FM reconstruction, and the worst case NICO reconstruction is on par with the average FM performance. We also note that the number of false positives and false negatives in a reconstruction using either scheme tend to be roughly equal (each constituting half of the edge symmetric difference error).

Figure 6 shows statistics for the number of edges in the reconstructed networks. There is an interesting correlation between the number of edges and reconstruction accuracy in this example. As seen above, the typical NICO reconstruction is more accurate, in terms of edge errors, than a FM reconstruction. NICO also consistently returns a sparser estimate. The median number of links in a NICO reconstruction is 1329, whereas the median number of links in a FM reconstruction
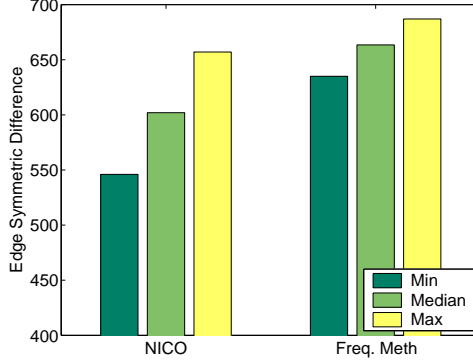
33

Figure 5: Edge symmetric difference error comparison of NICO and FM on Internet data. The reported values come from 50 random initializations of each algorithm.
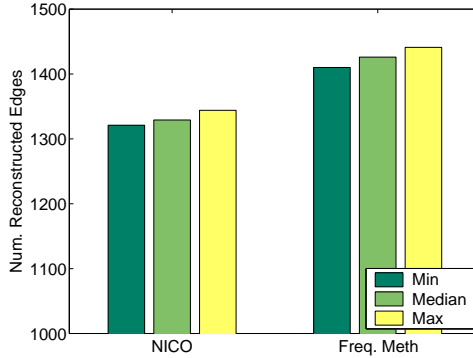

Figure 6: Number of edges in networks reconstructed using each method. The median number of edges per reconstruction is 1329 for NICO and 1426 for FM. The true network has 1317 edges, and so it appears that NICO does a better job of capturing the complexity of the true network.

is 1426. There are 1317 edges in the true network, so in this sense the NICO reconstructions more accurately reflect the inherent level of complexity in the true network.

Marginal log-likelihood values for each of the 50 NICO estimates are depicted in Figure 7. The marginal log-likelihood, given by (4), is the cost function being optimized by the EM algorithm. In contrast to the experiments with simulated data reported above, there is no exact correlation between higher marginal likelihood values and lower edge symmetric difference error for this example. The topology with the highest likelihood value results in an edge symmetric difference error of 627. This is better than the clairvoyant best FM error, but only average for NICO. The three repetitions which returned a topology with the lowest symmetric difference error had the next highest likelihood value, as indicated by the three hollow circles in the figure. The dashed line shows the
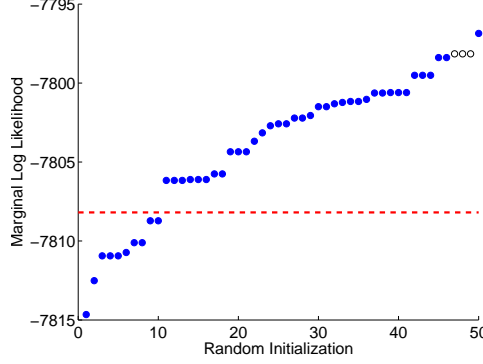
Figure 7: Marginal log likelihood values for different random initializations of NICO, sorted in ascending order. The three hollow circles correspond to the solutions which achieve the lowest edge symmetric difference error of all NICO trials. The red line shows the marginal log likelihood value computed using the true path orders to estimate a Markov transition matrix. Most NICO solutions have higher marginal log-likelihood than the true topology, suggesting that our generative model does not accurately describe Internet topology data.

likelihood value based on a transition matrix estimated using the true path orders as measured by `traceroute`. Notice that the majority of the NICO solutions have a higher marginal likelihood than the true topology. This suggests that our generative model may not be the best match for Internet topology data. Still the overall performance of our algorithm is encouraging.

7 Discussion and Ongoing Research

This paper presents a novel approach to network inference from co-occurrence observations. A co-occurrence observation reflects which vertices are activated by a particular transmission through the network, but not the order in which they are activated. We model transmission paths as a random walks on the underlying graph structure. Co-occurrence observations are modelled as i.i.d. samples of the random walk subjected to a random permutation which accounts for the lack of observed path order. Treating the random permutations as latent variables we derive an *expectation-maximization* (EM) algorithm for efficiently computing maximum likelihood or maximum *a posteriori* estimates of the random walk parameters (initial state distribution and transition matrix).

The complexity of the EM algorithm is dominated by the E-step calculation which is exponential in the length of the longest transmission path. In order to handle large networks, we describe fast

approximation methods based on importance sampling and Monte Carlo techniques. We derive concentration-style bounds on the accuracy of the Monte Carlo approximation. These bounds prescribe how many importance samples must be used to ensure a monotonic increase in the log-likelihood, thereby guaranteeing convergence of the algorithm with high probability. The resulting Monte Carlo EM computational complexity only depends polynomially on the length of the longest path.

To obtain a network reconstruction, we determine the most likely order for each co-occurrence observation according to the Markov chain parameter estimates, and then insert edges in the graph based on these ordered transmission paths. This procedure always produces a feasible reconstruction. The parameter estimates produced by the EM algorithm may be useful for other tasks such as guiding an expert to alternative reconstructions by assigning likelihoods to different permutations, or predicting unobserved paths through the network as in [12]. One could also analyze properties of an ensemble of solutions obtained by running the EM algorithm from different initializations, and then posit a new set of experiments to be conducted based on this analysis.

The transition matrix parameter $A_{i,j}$ can be interpreted as estimates of the probability that a transmission will be passed from vertex i to j, conditioned on the path reaching i; that is, $A_{i,j} = P[Z_{k+1} = j | Z_k = i]$. In particular, they *are not* estimates of the probability of a link existing from i to j. Since $\mathbf{A}$ is a stochastic matrix, each row must sum to 1, and so if vertex i is connected to many other nodes then the unit mass is being spread over more entries. We can obtain joint probabilities, $P[Z_k = i, Z_{k+1} = j]$, via Bayes theorem,

$$P[Z_{k+1} = j | Z_k = i] = \frac{P[Z_k = i, Z_{k+1} = j]}{P[Z_k = i]},$$

where $P[Z_k = i]$ is the stationary distribution of the chain (not necessarily equal to the initial state distribution). These joint probabilities (appropriately scaled versions of the transition matrix entries) more accurately reflect the likelihood of there being an edge from i to j, based on our estimates.

Our future work involves extending and generalizing both algorithmic and theoretical aspects of this work. In our experiments we found that our current model leads to reasonable Internet

reconstructions, but we feel there is room for improvement. For example, the structure of Internet paths may depend strongly on the destination of the traffic. We are currently investigating models based on "mixtures of random walks" to account for this added level of dependency.

Co-occurrence observations naturally arise from transmission *paths* in communication network applications and, to a degree, in biological, social, and brain networks as well. However the physical mechanisms driving interactions in the latter three applications may also correspond to more general connected subgraph structures such as trees or directed acyclic graphs. Extending our methods in this fashion is easily accomplished in theory, however the computational complexity may be significantly increased when more general structures are considered.

In this paper we have also restricted our attention to noise-free observations. We are also interested in extending our algorithm to handle the case where observations reflect a soft probability that a given vertex occurred in the path rather than hard, "active" or "not active", binary observations. This extension is relevant in many applications including the inference of signal transduction networks (in systems biology) where co-occurrence observations are themselves the result of inference procedures run on experimental data.

A   Proof of Theorem 1

There are two main steps in the proof of Theorem 1. First, we derive a concentration inequality for the importance sample approximations, $\widehat{\alpha}_{t',t''}^{(m)}$ and $\widehat{r}_{1,t'}^{(m)}$. Then we use the inequality to construct a bound for $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\boldsymbol{\theta})$.

Recall the expressions (22) and (23) for importance sample approximations calculated in the Monte Carlo E-step. Both are of the general form $\widehat{\mu}_L = \frac{\sum_{i=1}^{L} Z(\mathbf{r}^i) X(\mathbf{r}^i)}{\sum_{i=1}^{L} Z(\mathbf{r}^i)}$, where $Z : \Psi_N \to [0, b]$ and $X : \Psi_N \to \{0, 1\}$, and they are approximating $\mu = \sum_{\mathbf{r} \in \Psi_N} X(\mathbf{r}) P[\mathbf{r}|\mathbf{x}, \mathbf{A}, \boldsymbol{\pi}]$. Note that $\mathbb{E}[\widehat{\mu}_L] \neq \mu$, so standard concentration results such as Hoeffding's inequality or McDiarmid's bounded-differences

inequality do not directly apply; *e.g.*, consider the case $L = 1$:

$$\mathbb{E}\left[\frac{Z(\mathbf{r}^1)X(\mathbf{r}^1)}{Z(\mathbf{r}^1)}\right] = \sum_{\mathbf{r}\in\Psi_N} X(\mathbf{r})R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] \tag{41}$$

$$\neq \sum_{\mathbf{r}\in\Psi_N} X(\mathbf{r})P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]. \tag{42}$$

We can, however, show that $\widehat{\mu}_L$ yields an asymptotically consistent estimate of μ. Observe that

$$\mathbb{E}[Z(\mathbf{r}^i)] = \sum_{\mathbf{r}\in\Psi_N} \frac{P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}{R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] \tag{43}$$

$$= 1, \tag{44}$$

since P is a probability distribution on Ψ_N, and

$$\mathbb{E}[Z(\mathbf{r}^i)X(\mathbf{r}^i)] = \sum_{\mathbf{r}\in\Psi_N} \frac{P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}{R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}]}X(\mathbf{r})R[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] \tag{45}$$

$$= \sum_{\mathbf{r}\in\Psi_N} X(\mathbf{r})P[\mathbf{r}|\mathbf{x},\mathbf{A},\boldsymbol{\pi}] \tag{46}$$

$$= \mu. \tag{47}$$

It follows from the strong law of large numbers that $\widehat{\mu}_L \to \mu$ as $L \to \infty$.

The following finite-sample concentration inequality demonstrates that the approximation error, $\widehat{\mu}_N - \mu$, decays exponentially in the number of importance samples, L.

Proposition 1. *Let $\{(X_i, Z_i)\}$ be a sequence of independent and identically distributed random variables with $X_i \in \{0,1\}$ and $Z_i \in [0,b]$. Assume that $\mathbb{E}[Z_i] = 1$ and $\mathbb{E}[Z_iX_i] = \mu$, and set $\widehat{\mu}_L = \frac{\sum_{i=1}^L Z_iX_i}{\sum_{i=1}^L Z_i}$. Then with probability greater than $1 - \delta$,*

$$\widehat{\mu}_L - \mu < \sqrt{\frac{2b^2 \log\frac{2}{\delta}}{L}}. \tag{48}$$

Proof. From the definitions of Z_i and X_i, $Z_iX_i \in [0,b]$. Applying Hoeffding's inequality [10] yields

that for any $t > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i X_i - L\mu \geq Lt\right) \leq e^{-2Lt^2/b^2}, \tag{49}$$

and for any $t > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i - L \leq -Lt\right) \leq e^{-2Lt^2/b^2}. \tag{50}$$

Define the event, $E_t = \left\{\sum_{i=1}^{L} Z_i X_i - L\mu \geq Lt\right\} \bigcup \left\{\sum_{i=1}^{L} Z_i - L \leq -Lt\right\}$. By the union bound, $\Pr(E_t) \leq 2e^{-2Lt^2/b^2}$ for any $t > 0$. The complement of E_t implies that for $t \in (0,1)$,

$$\widehat{\mu}_L - \mu = \frac{\sum_{i=1}^{L} Z_i X_i - L\mu}{\sum_{i=1}^{L} Z_i} + \frac{L\mu}{\sum_{i=1}^{L} Z_i} - \mu \tag{51}$$

$$< \frac{Lt}{L(1-t)} + \frac{L\mu}{L(1-t)} - \mu \tag{52}$$

$$= \frac{t(1+\mu)}{1-t}. \tag{53}$$

It follows that $\left\{\widehat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\right\} \subseteq E_t$, and so $\Pr\left(\widehat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\right) \leq \Pr(E_t) \leq 2e^{-2Lt^2/b^2}$. Since $\widehat{\mu}_L \leq 1$, if $\frac{t(1+\mu)}{1-t} + \mu > 1$ then $\Pr\left(\widehat{\mu}_L - \mu \geq \frac{t(1+\mu)}{1-t}\right) = 0$, and the proposition holds trivially. Thus, without loss of generality we consider the case $\frac{t(1+\mu)}{1-t} + \mu \leq 1$, or equivalently, $t \leq (1-\mu)/2$. This restriction on t implies $\frac{t(1+\mu)}{1-t} \leq 2t$, and so we have $\Pr\left(\widehat{\mu}_L - \mu < 2t\right) > 1 - 2e^{-2Lt^2/b^2}$. Set $\delta = 2e^{-2Lt^2/b^2}$ to obtain the desired result. $\qquad\square$

We apply Proposition 1 to the Monte Carlo approximations $\{\widehat{\alpha}_{t',t''}^{(m)}\}$ and $\{\widehat{r}_{1,t'}^{(m)}\}$ as follows. Recall that the Monte Carlo weights are bounded according to $z_i \in [0, b_m]$, with b_m as defined in (38). Define

$$B_{\delta',L_m}^m = \left(\bigcup_{\substack{t',t''=1 \\ t' \neq t''}}^{N_m} \left\{\widehat{\alpha}_{t',t''}^{(m)} - \bar{\alpha}_{t',t''}^{(m)} \geq \sqrt{\frac{2b_m^2 \log \frac{2}{\delta'}}{L_m}}\right\}\right) \bigcup \left(\bigcup_{t'=1}^{N_m} \left\{\widehat{r}_{1,t'}^{(m)} - \bar{r}_{1,t'}^{(m)} \geq \sqrt{\frac{2b_m^2 \log \frac{2}{\delta'}}{L_m}}\right\}\right).$$

This is a union over $2\binom{N_m}{2} + N_m = N_m^2$ events, each of which holds with probability at most δ'

according to Proposition 1. By the union bound it follows that $\Pr(B^m_{\delta',L_m}) \leq N_m^2 \delta'$. Next, define

$$C^m_{\delta',L_m} = \left\{ \sum_{t',t''=1}^{N_m} \left(\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''} \right) + \sum_{t'=1}^{N_m} \left(\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'} \right) \geq N_m^2 \sqrt{\frac{2b_m^2 \log \frac{2}{\delta'}}{L_m}} \right\}, \tag{54}$$

and observe that $C^m_{\delta',L_m} \subseteq B^m_{\delta',L_m}$, therefore $\Pr(C^m_{\delta',L_m}) \leq \Pr(B^m_{\delta',L_m}) \leq N_m^2 \delta'$. Let $\delta'' = N_m^2 \delta'$ and let $L > 0$ be a value to be determined later. For each $m = 1, \ldots, T$, set

$$L_m = \frac{2L N_m^4 b_m^2 \log \frac{2N_m^2}{\delta''}}{\log \frac{1}{\delta''}}, \tag{55}$$

so that

$$N_m^2 \sqrt{\frac{2b_m^2 \log \frac{2}{\delta'}}{L_m}} = N_m^2 \sqrt{\frac{2b_m^2 \log \frac{2N_m^2}{\delta''}}{L_m}} = \sqrt{\frac{\log \frac{1}{\delta''}}{L}}. \tag{56}$$

Then with probability greater than $1 - \delta''$,

$$\sum_{t',t''=1}^{N_m} \left(\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''} \right) + \sum_{t'=1}^{N_m} \left(\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'} \right) < \sqrt{\frac{\log \frac{1}{\delta''}}{L}}. \tag{57}$$

Recall that $x^{(m)}_{t',i}$ are indicator variables satisfying $\sum_{i,j=1}^{|S|} x^{(m)}_{t'',i} x^{(m)}_{t',j} = 1$ and $\sum_{i=1}^{|S|} x^{(m)}_{t',i} = 1$. Multiplying each term in (57) by the appropriate sum of indicators, rearranging terms, and recalling that importance sample estimates for different observations are statistically independent, we have that with probability greater than $(1 - \delta'')^T$,

$$\bigcap_{m=1}^{T} \left\{ \sum_{t',t''=1}^{N_m} \sum_{i,j=1}^{|S|} \left(\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''} \right) x^{(m)}_{t'',i} x^{(m)}_{t',j} + \sum_{t'=1}^{N_m} \sum_{i=1}^{|S|} \left(\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'} \right) x^{(m)}_{t',i} < \sqrt{\frac{\log \frac{1}{\delta''}}{L}} \right\}, \tag{58}$$

which implies that with probability greater than $(1 - \delta'')^T$,

$$\sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \sum_{i,j=1}^{|S|} \left(\widehat{\alpha}^{(m)}_{t',t''} - \bar{\alpha}^{(m)}_{t',t''} \right) x^{(m)}_{t'',i} x^{(m)}_{t',j} + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \sum_{i=1}^{|S|} \left(\widehat{r}^{(m)}_{1,t'} - \bar{r}^{(m)}_{1,t'} \right) x^{(m)}_{t',i} < T \sqrt{\frac{\log \frac{1}{\delta''}}{L}}. \tag{59}$$

Finally, set $1 - \delta = (1 - \delta'')^T$ and multiply through by $|\log \theta_{\min}| > 0$. Then with probability greater

than $1 - \delta$,

$$\sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \sum_{i,j=1}^{|S|} \left(\widehat{\alpha}_{t',t''}^{(m)} - \bar{\alpha}_{t',t''}^{(m)} \right) x_{t'',i}^{(m)} x_{t',j}^{(m)} \, |\log \theta_{\min}| + \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \sum_{i=1}^{|S|} \left(\widehat{r}_{1,t'}^{(m)} - \bar{r}_{1,t'}^{(m)} \right) x_{t',i}^{(m)} \, |\log \theta_{\min}|$$

$$< \quad T \, |\log \theta_{\min}| \sqrt{\frac{-\log \left(1 - (1 - \delta)^{1/T} \right)}{L}}. \tag{60}$$

To complete the proof, observe that

$$\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) \quad = \quad \sum_{m=1}^{T} \sum_{i,j=1}^{|S|} \sum_{t',t''}^{N_m} \left(\widehat{\alpha}_{t',t''}^{(m)} - \bar{\alpha}_{t',t''}^{(m)} \right) x_{t'',i}^{(m)} x_{t',j}^{(m)} \left(\log \widehat{A}_{i,j} - \log A_{i,j}' \right)$$

$$+ \sum_{m=1}^{T} \sum_{i=1}^{|S|} \sum_{t'=1}^{N_m} \left(\widehat{r}_{1,t'}^{(m)} - \bar{r}_{1,t'}^{(m)} \right) x_{t',i}^{(m)} \left(\log \widehat{\pi}_i - \log \pi_i' \right). \tag{61}$$

By assumption, $\theta_{\min} \le \widehat{A}_{i,j}, A_{i,j}' \le 1$ for each $(i, j) \in S^2$. It follows that

$$\log \widehat{A}_{i,j} - \log A_{i,j}' \quad \le \quad -\log \theta_{\min} \quad = \quad |\log \theta_{\min}|. \tag{62}$$

Similarly, $\log \widehat{\pi}_i - \log \pi_i' \le |\log \theta_{\min}|$ for each $i \in S$. Apply these bounds in (61) to find that the right hand side of (61) is no greater than the left hand side of (60). Set

$$\epsilon \quad = \quad T \, |\log \theta_{\min}| \sqrt{\frac{\log \frac{1}{1 - (1 - \delta)^{1/T}}}{L}}. \tag{63}$$

Then $\widehat{\Delta}(\widehat{\boldsymbol{\theta}}) - \Delta(\widehat{\boldsymbol{\theta}}) < \epsilon$ with probability greater than $1 - \delta$. Solve for L in (63) and plug the resulting value back into (55) with $\delta'' = 1 - (1 - \delta)^{1/T}$ to obtain the desired result.

B   Proof of Theorem 2

To prove Theorem 2 we will show that $\Delta(\widehat{\boldsymbol{\theta}}) > (1 - \epsilon) \, \Delta(\boldsymbol{\theta}^*)$ with high probability, but first we need two preliminary results. We begin by deriving concentration inequalities for the Monte Carlo sufficient statistics. Then we use these bounds to show that the corresponding M-step parameter estimates, $\widehat{A}_{i,j}$ and $\widehat{\pi}_i$ concentrate about their asymptotic means, $A_{i,j}^*$ and π_i^*. From there we

construct the desired bound for $\Delta(\widehat{\boldsymbol{\theta}})$, which implies the theorem since $\Delta(\boldsymbol{\theta}^*) \geq 0$ by definition.

The proof of Theorem 1 makes use of *additive* concentration inequalities, bounding the probability of deviations of the form $\widehat{\mu}_L - \mu \geq t$. In this proof we make use of *relative* concentration inequalities to ensure that $\widehat{\mu}_L > (1 + \epsilon)\mu$ with high probability.

Proposition 2. *Let* $\{(X_i, Z_i)\}$ *be a sequence of independent and identically distributed random variables with* $X_i \in \{0, 1\}$ *and* $Z_i \in [0, b]$. *Assume that* $\mathbb{E}[Z_i] = 1$ *and* $\mathbb{E}[Z_i X_i] = \mu$, *and set* $\widehat{\mu}_L = \frac{\sum_{i=1}^{L} Z_i X_i}{\sum_{i=1}^{L} Z_i}$, *as before. Then with probability greater than* $1 - \delta$,

$$\widehat{\mu}_L \;<\; \left(1 + \sqrt{\frac{27b\log\frac{2}{\delta}}{L\mu}}\right)\mu,$$

and with probability greater than $1 - \delta$,

$$\widehat{\mu}_L \;>\; \left(1 - \sqrt{\frac{27b\log\frac{2}{\delta}}{L\mu}}\right)\mu.$$

Proof. Since $X_i \in \{0,1\}$ and $Z_i \in [0, b]$, $Z_i X_i \in [0, b]$ also. Applying the relative form of Hoeffding's inequality (see, *e.g.*, Theorem 2.3 in [17]), we have that for any $\beta > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i X_i \geq (1 + \beta)L\mu\right) \;\leq\; \exp\left\{\frac{-L\mu\beta^2}{2b(1 + \beta/3)}\right\}. \tag{64}$$

If $\beta \leq 1$ then $2(1 + \beta/3) < 3$, and so for $\beta \in (0, 1]$,

$$\Pr\left(\sum_{i=1}^{L} Z_i X_i \geq (1 + \beta)L\mu\right) \;\leq\; \exp\left\{\frac{-L\mu\beta^2}{3b}\right\}, \tag{65}$$

which suffices for our application. Also, for any $\gamma > 0$,

$$\Pr\left(\sum_{i=1}^{L} Z_i \leq (1 - \gamma)L\right) \;\leq\; \exp\left\{\frac{-L\gamma^2}{2b}\right\}. \tag{66}$$

Suppose the events

$$\left\{\sum_{i=1}^{L} Z_i X_i < (1+\beta) L\mu\right\} \quad \text{and} \quad \left\{\sum_{i=1}^{L} Z_i > (1-\gamma)L\right\} \tag{67}$$

occur simultaneously. Then for $0 < \gamma < 1$,

$$\widehat{\mu}_L > \left(\frac{1+\beta}{1-\gamma}\right)\mu. \tag{68}$$

Since we will apply the union bound, we balance the exponential rates in (65) and (66) by setting $\gamma = \beta\sqrt{\frac{2}{3}\mu} < 1$. Solving

$$\frac{1+\beta}{1-\gamma} = \frac{1+\beta}{1-\beta\sqrt{\frac{2}{3}\mu}} = 1 + \epsilon \tag{69}$$

for β in terms of ϵ leads to

$$\beta = \frac{\epsilon}{1 + \sqrt{\frac{2}{3}\mu} + \epsilon\sqrt{\frac{2}{3}\mu}}. \tag{70}$$

In order to ensure that $\beta \leq 1$ we restrict

$$\epsilon \leq \frac{1+1}{1-\sqrt{\frac{2}{3}\mu}} - 1 \tag{71}$$

$$= \frac{1+\sqrt{\frac{2}{3}\mu}}{1-\sqrt{\frac{2}{3}\mu}}. \tag{72}$$

Note that the right hand side of the expression above is at least 1 for all $\mu \in [0,1]$. Apply the union bound with the complements of the events in (67) using (70) in the exponent, and observe that $1 + \sqrt{\frac{2}{3}\mu} + \epsilon\sqrt{\frac{2}{3}\mu} \leq 3$ for all $\mu \in [0,1]$ and $\epsilon \in (0,1)$ to find that $\Pr(\widehat{\mu}_L \leq (1+\epsilon)\mu) \leq 2e^{-L\mu\epsilon^2/27b^2}$. Set $\delta = 2e^{-L\mu\epsilon^2/27b^2}$ to obtain the first claim. The proof of the second claim follows a similar sequence of steps. See [20] for the full details. $\square$

Application of the union bound yields the following.

Corollary 1. *With probability greater than $1 - \delta$,*

$$\left(1 - \sqrt{\frac{27b\log\frac{4}{\delta}}{L\mu}}\right)\mu \; < \; \widehat{\mu}_L \; < \; \left(1 + \sqrt{\frac{27b\log\frac{4}{\delta}}{L\mu}}\right)\mu. \tag{73}$$

Next we apply Corollary 1 to the Monte Carlo approximations, $\{\widehat{r}_{1,t'}^{(m)}\}$ and $\{\widehat{\alpha}_{t',t''}^{(m)}\}$ towards showing that $\widehat{A}_{i,j}$ and $\widehat{\pi}_i$ do not deviate too greatly from $A_{i,j}^*$ and π_i^*. Recall the exact M-step expressions for π_i^* and $A_{i,j}^*$ given by (18). The corresponding expressions for $\widehat{\pi}_i$ and $\widehat{A}_{i,j}$ are found by replacing each $\bar{r}_{1,t'}^{(m)}$ and $\bar{\alpha}_{t',t''}^{(m)}$ with $\widehat{r}_{1,t'}^{(m)}$ and $\widehat{\alpha}_{t',t''}^{(m)}$. By appropriately bounding the numerators and denominators of $\widehat{A}_{i,j}$ and $\widehat{\pi}_i$ we obtain the following result.

Proposition 3. *Let $L > 0$ and $\delta > 0$ be given. Assume that there exists $\lambda > 0$ such that $\bar{r}_{1,t'}^{(m)} \geq \lambda$ and $\bar{\alpha}_{t',t''}^{(m)} \geq \lambda$ for all $m = 1,\ldots,T$ and $t',t'' = 1,\ldots,N_m$. If at least $L_m \geq \frac{27b_m L}{\lambda}$ importance samples are used, then with probability at least $1 - (\sum_{m=1}^{T} N_m^2)\delta$,*

$$\left(\bigcap_{i,j=1}^{|S|}\left\{\widehat{A}_{i,j} > \left(\frac{1 - \sqrt{\frac{\log\frac{4}{\delta}}{L}}}{1 + \sqrt{\frac{\log\frac{4}{\delta}}{L}}}\right)A_{i,j}^*\right\}\right) \cap \left(\bigcap_{i=1}^{|S|}\left\{\widehat{\pi}_i > \left(\frac{1 - \sqrt{\frac{\log\frac{4}{\delta}}{L}}}{1 + \sqrt{\frac{\log\frac{4}{\delta}}{L}}}\right)\pi_i^*\right\}\right). \tag{74}$$

Proof. First recall that there are $2\binom{N_m}{2} + N_m = N_m^2$ sufficient statistics associated with the mth observation: one $\alpha_{t',t''}^{(m)}$ for each of the $2\binom{N_m}{2}$ possible transitions and one $r_{1,t'}^{(m)}$ for each possible initial state. Then, in total there are $\sum_{m=1}^{T} N_m^2$ sufficient statistics to calculate in the E-step. Applying the union bound in conjunction with (73) we have that with probability greater than $1 - (\sum_{m=1}^{T} N_m^2)\delta$,

$$\bar{\alpha}_{t',t''}^{(m)} - \sqrt{\frac{27b_m\bar{\alpha}_{t',t''}^{(m)}\log\frac{4}{\delta}}{L_m}} \; < \; \widehat{\alpha}_{t',t''}^{(m)} \; < \; \bar{\alpha}_{t',t''}^{(m)} + \sqrt{\frac{27b_m\bar{\alpha}_{t',t''}^{(m)}\log\frac{4}{\delta}}{L_m}}, \tag{75}$$

for all $m = 1,\ldots,T$ and $t',t'' = 1,\ldots,N_m$, and

$$\bar{r}_{1,t'}^{(m)} - \sqrt{\frac{27b_m\bar{r}_{1,t'}^{(m)}\log\frac{4}{\delta}}{L_m}} \; < \; \widehat{r}_{1,t'}^{(m)} \; < \; \bar{r}_{1,t'}^{(m)} + \sqrt{\frac{27b_m\bar{r}_{1,t'}^{(m)}\log\frac{4}{\delta}}{L_m}}, \tag{76}$$

for all $m = 1,\ldots,T$ and $t' = 1,\ldots,N_m$. Based on the assumption that $\bar{\alpha}_{t',t''}^{(m)} \geq \lambda$ and $\bar{r}_{1,t'}^{(m)} \geq \lambda$,

44

taking $L_m \geq 27 b_m L / \lambda$ guarantees that

$$L_m \quad \geq \quad \max \left(\max_{t', t''=1, \ldots, N_m} \frac{27 b_m L}{\bar{\alpha}_{t', t''}^{(m)}}; \quad \max_{t'=1, \ldots, N_m} \frac{27 b_m L}{\bar{r}_{1, t'}^{(m)}} \right). \tag{77}$$

Then with probability greater than $1 - (\sum_{m=1}^{T} N_m^2) \delta$,

$$\left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}} \right) \bar{\alpha}_{t', t''}^{(m)} \quad < \quad \widehat{\alpha}_{t', t''}^{(m)} \quad < \quad \left(1 + \sqrt{\frac{\log \frac{4}{\delta}}{L}} \right) \bar{\alpha}_{t', t''}^{(m)}, \tag{78}$$

for all $m = 1, \ldots, T$ and $t', t'' = 1, \ldots, N_m$, and

$$\left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}} \right) \bar{r}_{1, t'}^{(m)} \quad < \quad \widehat{r}_{1, t'}^{(m)} \quad < \quad \left(1 + \sqrt{\frac{\log \frac{4}{\delta}}{L}} \right) \bar{r}_{1, t'}^{(m)}, \tag{79}$$

for all $m = 1, \ldots, T$ and $t' = 1, \ldots, N_m$.

Equation (78) implies that for each $(i, j) \in S^2$,

$$\sum_{m=1}^{T} \sum_{t', t''=1}^{N_m} \widehat{\alpha}_{t', t''}^{(m)} x_{t'', i}^{(m)} x_{t', j}^{(m)} \quad > \quad \left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}} \right) \sum_{m=1}^{T} \sum_{t', t''=1}^{N_m} \bar{\alpha}_{t', t''}^{(m)} x_{t'', i}^{(m)} x_{t', j}^{(m)},$$

and for each $i \in S$,

$$\sum_{k=1}^{|S|} \sum_{m=1}^{T} \sum_{t', t''}^{(m)} \widehat{\alpha}_{t', t''}^{(m)} x_{t'', i}^{(m)} x_{t', k}^{(m)} \quad < \quad \left(1 + \sqrt{\frac{\log \frac{4}{\delta}}{L}} \right) \sum_{k=1}^{|S|} \sum_{m=1}^{T} \sum_{t', t''=1}^{N_m} \bar{\alpha}_{t', t''}^{(m)} x_{t'', i}^{(m)} x_{t', k}^{(m)}.$$

Taking the ratio of these two expressions yields the desired result for $\widehat{A}_{i,j}$ and $A_{i,j}^*$.

Similarly, (79) implies that for each i,

$$\sum_{m=1}^{T} \sum_{t'=1}^{N_m} \widehat{r}_{1, t'}^{(m)} x_{t', i}^{(m)} \quad > \quad \left(1 - \sqrt{\frac{\log \frac{4}{\delta}}{L}} \right) \sum_{m=1}^{T} \sum_{t'} \bar{r}_{1, t'}^{(m)} x_{t', i}^{(m)}, \tag{80}$$

and for each i,

$$\sum_{m=1}^{T} \sum_{t'=1}^{N_m} \widehat{r}_{1,t'}^{(m)} x_{t',i}^{(m)} \;<\; \left(1 + \sqrt{\frac{\log \frac{4}{\delta}}{L}}\right) \sum_{m=1}^{T} \sum_{t'}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)}. \tag{81}$$

Taking the ratio of these two expressions yields the desired result for $\widehat{\pi}_i$ and π_i^*. $\qquad\square$

The remainder of the proof of Theorem 2 is now fairly straightforward. Let $\delta > 0$ be the value given in the statement of Theorem 2. Monotonicity of the logarithm in conjunction with Proposition 3 implies that with probability greater than $1 - \delta$, for all $i, j \in S$,

$$\log \widehat{A}_{i,j} \;>\; \log A_{i,j}^* + \log \left(\frac{1 - \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}{1 + \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}\right), \tag{82}$$

$$\log \widehat{\pi}_i \;>\; \log \pi_i^* + \log \left(\frac{1 - \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}{1 + \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}\right). \tag{83}$$

Multiply through by either $\sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} > 0$ or $\sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t',i}^{(m)} > 0$ as appropriate, and sum over i and j to obtain

$$
\begin{aligned}
Q(\widehat{\boldsymbol{\theta}}; \boldsymbol{\theta}') \;>\; & Q(\boldsymbol{\theta}^*; \boldsymbol{\theta}') \\
& + \left(\sum_{i,j=1}^{|S|} \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} + \sum_{i=1}^{|S|} \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)}\right) \log \left(\frac{1 - \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}{1 + \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}\right).
\end{aligned}
\tag{84}
$$

By the definitions of $x_{t',i}^{(m)}$, $\bar{r}_{1,t'}^{(m)}$, and $\bar{\alpha}_{t',t''}^{(m)}$ given in Section 3 we have $\sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} = 1$, $\sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} = N_m - 1$, $\sum_{i,j=1}^{|S|} x_{t'',i}^{(m)} x_{t',j}^{(m)} = 1$, and $\sum_{i=1}^{|S|} x_{t',i}^{(m)} = 1$. It follows that

$$\sum_{i,j=1}^{|S|} \sum_{m=1}^{T} \sum_{t',t''=1}^{N_m} \bar{\alpha}_{t',t''}^{(m)} x_{t'',i}^{(m)} x_{t',j}^{(m)} + \sum_{i=1}^{|S|} \sum_{m=1}^{T} \sum_{t'=1}^{N_m} \bar{r}_{1,t'}^{(m)} x_{t',i}^{(m)} \;=\; \sum_{m=1}^{T} N_m. \tag{85}$$

46

Subtract $Q(\boldsymbol{\theta}'; \boldsymbol{\theta}')$ from both sides of (84) to obtain that with probability greater than $1 - \delta$,

$$\Delta(\widehat{\boldsymbol{\theta}}) \; > \; \Delta(\boldsymbol{\theta}^*) + \left(\sum_{m=1}^{T} N_m\right) \log\left(\frac{1 - \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}{1 + \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}\right). \tag{86}$$

Let $\epsilon > 0$ be the value given in the statement of Theorem 2 and set

$$\left(\sum_{m=1}^{T} N_m\right) \log\left(\frac{1 - \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}{1 + \sqrt{\frac{\log 4 \sum_{m=1}^{T} N_m^2 - \log \delta}{L}}}\right) \; = \; -\epsilon\Delta(\boldsymbol{\theta}^*). \tag{87}$$

Solving for L yields

$$L \; = \; \left(\frac{1 + \exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m}\right\}}{1 - \exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m}\right\}}\right)^2 \log\left(\frac{4 \sum_{m=1}^{T} N_m^2}{\delta}\right). \tag{88}$$

Recall the well known inequality: $u \geq \log(1 + u)$ for $u \geq 0$. Putting $u = \frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m} \geq 0$ leads to

$$\frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m} \; \geq \; 1 + \frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m}. \tag{89}$$

Take the exponential, which is a monotonic transformation, and then invert the resulting expression to obtain

$$\exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m}\right\} \; \leq \; \left(1 + \frac{\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m}\right)^{-1}. \tag{90}$$

It follows that

$$\frac{1 + \exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m}\right\}}{1 - \exp\left\{\frac{-\epsilon\Delta(\boldsymbol{\theta}^*)}{\sum_{m=1}^{T} N_m}\right\}} \; \leq \; \frac{2 \sum_{m=1}^{T} N_m + \epsilon\Delta(\boldsymbol{\theta}^*)}{\epsilon\Delta(\boldsymbol{\theta}^*)}. \tag{91}$$

Using this last result in (88), together with the choice of L_m from Proposition 3, we find that if we

use

$$L_m \;\;=\;\; \frac{27 b_m}{\lambda} \left(\frac{2 \sum_{m=1}^{T} N_m + \epsilon \Delta(\boldsymbol{\theta}^*)}{\epsilon \Delta(\boldsymbol{\theta}^*)} \right)^2 \log \left(\frac{4 \sum_{m=1}^{T} N_m^2}{\delta} \right) \tag{92}$$

importance samples for the mth observation in the Monte Carlo E-step, then $\Delta(\widehat{\boldsymbol{\theta}}) \geq (1 - \epsilon) \Delta(\boldsymbol{\theta}^*)$ with probability greater than $1 - \delta$. Since $\Delta(\boldsymbol{\theta}^*) \geq 0$ by definition we may take $\epsilon = 1$. Then $\Delta(\widehat{\boldsymbol{\theta}}^*) \geq 0$ with probability greater than $1 - \delta$.

References

[1] *International Workshop on Brain Connectivity*, 2005. `http://www.ccs.fau.edu/~bc2005/welcome.html`.

[2] J. Bernardo and A. Smith. *Bayesian Theory*. John Wiley & Sons, 1994.

[3] J. G. Booth, J. P. Hobert, and W. S. Jank. A survey of Monte Carlo algorithms for maximizing the likelihood of a two-stage hierarchical model. *Statistical Modelling*, 1:333–349, 2001.

[4] R. A. Boyles. On the convergence of the EM algorithm. *Journal of the Royal Statistical Society B*, 45(1):47–50, 1983.

[5] B. S. Caffo, W. Jank, and G. L. Jones. Ascent-based Monte Carlo EM. *Journal of the Royal Statistical Society B*, 67(2):235–252, 2005.

[6] M. Coates, A. O. Hero, R. Nowak, and B. Yu. Internet tomography. *IEEE Signal Processing Magazine*, 19(3):47–65, 2002.

[7] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, March 2002.

[8] N. Friedman and D. Koller. Being Bayesian about Bayesian network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125, 2003.

[9] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

[10] W. J. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:713–721, 1963.

[11] W. Jank. Stochastic variants of the EM algorithm: Monte Carlo, quasi-Monte Carlo and more. In *Proc. of the American Statistical Association*, Minneapolis, Minnesota, August 2005.

[12] D. Justice and A. O. Hero. Estimation of message source and destination from link intercepts. Submitted to *IEEE Trans. on Information Forensics and Security*, April 2005.

[13] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach. *Systems Biology in Practice: Concepts, Implementation and Application.* John Wiley and Sons, 2005.

[14] J. Kubica, A. Moore, D. Cohn, and J. Schneider. cGraph: A fast graph-based method for link analysis and queries. In *Proc. IJCAI Text-Mining and Link-Analysis Workshop*, Acapulco, Mexico, August 2003.

[15] J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer, 2001.

[16] Y. Liu and H. Zhao. A computational approach for ordering signal transduction pathway components from genomics and proteomics data. *BMC Bioinformatics*, 5(158), October 2004.

[17] C. McDiarmid. Concentration. In M. Habib, C. McDiarmid, J. Ramirez-Alfonsin, and B. Reed, editors, *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248. Springer-Verlag, New York, 1998.

[18] M. Newman, A. L. Barabasi, and D. J. Watts. *The Structure and Dynamics of Networks.* Princeton University Pres, 2006.

[19] B. O. Palsson. *Systems Biology: Properties of Reconstructed Networks.* Cambridge University Press, 2006.

[20] M. G. Rabbat, M. A. T. Figueiredo, and Robert D. Nowak. Network inference from co-occurrences. Technical report ECE-06-02, Department of Electrical and Computer Engineering, University of Wisconsin-Madison, April 2006.

[21] M. G. Rabbat, J. R. Treichler, S. L. Wood, and M. G. Larimore. Understanding the topology of a telephone network via internally-sensed network tomography. In *Proc. IEEE International Confernece on Acoustics, Speech, and Signal Processing*, volume 3, pages 977–980, Philadelphia, PA, March 2005.

[22] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Verlag, New York, 1999.

[23] O. Sporns, D. R. Chialvo, M. Kaiser, and C. C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Science*, 8(9), 2004.

[24] O. Sporns and G. Tononi. Classes of network connectivity and dynamics. *Complexity*, 7(1):28–38, 2002.

[25] S. Wasserman, K. Faust, D. Iacobucci, and M. Granovetter. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[26] G. C. G. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor mans data augmentation algorithms. *Journal of the American Statistical Association*, 85:699–704, 1990.

[27] C. F. J. Wu. On the convergence properties of the EM algorithm. *Annals of Statistics*, 11(1):95–103, 1983.

[28] D. Zhu, A. O. Hero, H. Cheng, R. Khanna, and A. Swaroop. Network constrained clustering for gene microarray data. *Bioinformatics*, 21(21):4014–4020, 2005.